G 6

```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
```

```
_$2

Val
---
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
```

```
**FILE**ID**INPUTOBJ
```

```
  IIIIII    NN      NN  PPPPPPPP    UU      UU  TTTTTTTTTT    000000    BBBBBBBB            JJ
  IIIIII    NN      NN  PPPPPPPP    UU      UU  TTTTTTTTTT    000000    BBBBBBBB            JJ
    II      NN      NN  PP      PP  UU      UU      TT      00      00  BB      BB          JJ
    II      NN      NN  PP      PP  UU      UU      TT      00      00  BB      BB          JJ
    II      NNNN    NN  PP      PP  UU      UU      TT      00      00  BB      BB          JJ
    II      NNNN    NN  PP      PP  UU      UU      TT      00      00  BB      BB          JJ
    II      NN  NN  NN  PPPPPPPP    UU      UU      TT      00      00  BBBBBBBB            JJ
    II      NN  NN  NN  PPPPPPPP    UU      UU      TT      00      00  BBBBBBBB            JJ
    II      NN    NNNN  PP          UU      UU      TT      00      00  BB      BB  JJ      JJ
    II      NN    NNNN  PP          UU      UU      TT      00      00  BB      BB  JJ      JJ
    II      NN      NN  PP          UU      UU      TT      00      00  BB      BB  JJ      JJ    ....
    II      NN      NN  PP          UU      UU      TT      00      00  BB      BB  JJ      JJ    ....
  IIIIII    NN      NN  PP          UUUUUUUUUU      TT        000000    BBBBBBBB      JJJJJJ      ....
  IIIIII    NN      NN  PP          UUUUUUUUUU      TT        000000    BBBBBBBB      JJJJJJ      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE lib_inputobj (
0                       LANGUAGE (BLISS32),
0                       IDENT = 'V04-000'
0                       ) =
0
1 BEGIN
1 !
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 !
1 !++
1 !
1 ! FACILITY:  Library command processor
1 !
1 ! ABSTRACT:
1 !
1 !       The VAX/VMS librarian is invoked by DCL to process the LIBRARY
1 !       command.  It utilizes the librarian procedure set to perform
1 !       the actual modifications to the library.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX native, user mode.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Benn Schreiber,       CREATION DATE:  12-June-1979
1 !
1 ! MODIFIED BY:
1 !
1 !       V02-008         RPG0048         Bob Grosso      11-Mar-1982
1 !               When symbol multiply defined in the same module,
1 !               disregard subsequent references.
1 !               Also fix up several places where $BYTEOFFSET should be used.
```

```
1 !
1 !          V02-007           RPG0047         Bob Grosso       02-Feb-1982
1 !                   Support for logging replace operations in history.
1 !
1 !          V02-006           RPG0046         Bob Grosso       21-Nov-1981
1 !                   Support new GSD records
1 !
1 !          V02-005           RPG0045         Bob Grosso       7-Aug-1981
1 !                   lib$gl_ctlmsk now a quadword
1 !
1 !          V02-004           RPG0036         Bob Grosso       25-Jun-1981
1 !                   Continue after a duplicate module.
1 !
1 !          V02-003           RPG0035         Bob Grosso       22-Apr-1981
1 !                   Record module names for update history.
1 !
1 !          V02-002 BLS0029          Benn Schreiber          23-Dec-1980
1 !                   Convert messages to message compiler.  Add library of
1 !                   shareable image symbol tables.
1 !--
```

```
1 %SBTTL  'Declarations';
1
1 LIBRARY
1           'SYS$LIBRARY:LIB.L32';          !System macro definitions
1 REQUIRE
1           'PREFIX';                   !SET OF GENERAL MACROS ETC
1 REQUIRE
1           'LIBDEF';                   !Librarian structure defs.
1 REQUIRE
1           'LBRDEF';                   !Library processor defs.
1
1 EXTERNAL
1     lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL),  !RMS STV from librarian
1     lib$gl_objmodix,              !Index number for module name index
1     lib$gl_objgsdix,              !index number for gsd symbols
1     lib$gl_recount,               !Count of records inserted
1     lib$al_rab : BBLOCK,          !Input file RAB
1     lib$gl_type,                  !Type of library opened
1     lib$gl_keysize,               !Max size of key
1     lib$gl_ctlmsk : BLOCK [2],    !Control flags
1     lib$gl_libfdb : REF BBLOCK,   !Pointer to library fdb
1     lib$gl_inpfdb : REF BBLOCK,   !Pointer to input file fdb
1     lib$gl_libctl;                !Library control index
1
1 FORWARD ROUTINE
1     prorec,                       !check sequence and copy record
1     copyrec,                      !copy record to object library
1     prohdr,                       !Routine to process module headers
1     protir,                       !Routine to process TIR records
1     progsd,                       !Routine to process gsd records
1     proeom,                       !   "        "        end of module
1     seqchk,                       !   "       verify correct sequence of obj records
1     propsectdef,                  !Process p-section definitions
1     symbols,                      !Process symbol definitions and references
1     entpnts,                      !Process entry point definitions
1     procedef,                     !Process procedure declarations
1     pro_epmw,                     !Process entry point definition with word psect
1     pro_idc,                      !Process random entity check
1     pro_env,                      !Process environment definition
1     pro_lsy,                      !Process local symbol definition/reference
1     pro_lepm,                     !Process local symbol entry point definition
1     pro_lpro,                     !Process local symbol procedure definition
1     pro_spsc,                     !Process shareable image psect definition
1     profile,                      !Read all records of file
1     finish_object,                !Do end of module processing
1     delsym,                       !Add symbol to delete symbol list
1     prosymbol;                    !Do all the work of symbol resolution
1
1 EXTERNAL ROUTINE
1     lib_get_mem,                                  !Allocate virtual memory
1     lib_get_zmem,                                 !Allocate zeroed virtual memory
1     lib_free_mem,                                 !and give it back
1     lib_log_op,                                   !Log operation on console
1     lib_log_upd,                                  !record module names for LUH
1     lbr$search : ADDRESSING_MODE (GENERAL),       !Search index for keys with RFA
1     lbr$delete_data : ADDRESSING_MODE (GENERAL),  !Delete data
1     lbr$put_record : ADDRESSING_MODE (GENERAL),   !Write record to library
```

```
        lbr$put_end     : ADDRESSING_MODE (GENERAL), !Terminated writing records
        lbr$lookup_key : ADDRESSING_MODE (GENERAL),  !Lookup key in library
        lbr$set_index : ADDRESSING_MODE (GENERAL),   !Set index number
        lbr$insert_key : ADDRESSING_MODE (GENERAL),  !Insert key
        lbr$set_module : ADDRESSING_MODE (GENERAL),  !Set module attributes
        lbr$replace_key : ADDRESSING_MODE (GENERAL), !Replace key
        lbr$delete_key : ADDRESSING_MODE (GENERAL),  !Delete key from library
        get_record;                                  !Get next input record

    EXTERNAL LITERAL
        lib$_notshrimg,                              !File not shareable image
        lib$_nosymbols,                              !No stb in shareable image
        lib$_reclng,                                 !Illegal record length
        lib$_rectyp,                                 !Illegal record type
        lib$_noeom,                                  !No eom record
        lib$_strlvl,                                 !Illegal structure level
        lib$_modnamlng,                              !Illegal module name length
        lib$_indexerr,                               !Index error
        lib$_inserted,                               !Module inserted
        lib$_replaced,                               !Module replaced
        lib$_dupmodule,                              !Duplicate module
        lib$_gsdtyp,                                 !Illegal gsd type
        lib$_spnamlng,                               !Illegal psect name length
        lib$_symnamlng,                              !Illegal symbol name length
        lib$_dupglobal,                              !Duplicate global
        lib$_comcod,                                 !Compilation errors in module
        lib$_mhderr,                                 !Module header error
        lib$_inserterr,                              !Insertion error
        lib$_delkeyerr,                              !Delete key error
        lib$_deldaterr,                              !Delete data error
        lib$_seqnce;                                 !Record sequence error

    OWN
        shrgsmatch,                                             !GSMATCH for shareable image
        operation,
        mhdseen,
        lnmseen,
        dupseen,                                                ! Record that a duplicate module is being processed
        gsdoffset,                                              !Offset into concatenated gsd record
        symbolstring          : REF VECTOR [,BYTE],             !Pointer to current symbol
        recdesc : BBLOCK [dsc$c_s_bln],                         !String descriptor for record
        lastrectyp,                                             !Type of the previous record
        currectyp    : INITIAL (obj$c_eom),                     !Type of the current record
        maxreclng    : INITIAL (obj$c_maxrecsiz),               !Maximum record length
        mod_name     : VECTOR [sym$c_maxlng+1, BYTE],           !Module name
        modulerfa    : BBLOCK [rfa$c_length],                   !RFA of module text
        oldmodrfa    : BBLOCK [rfa$c_length],                   !RFA of old module text
        replacing,                                              !Flag if replacing this module
        moduledesc : BBLOCK [dsc$c_s_bln] INITIAL               !String descriptor for module name
                         (0, mod_name [1]),
        moduledata   : VECTOR [sym$c_maxlng + 2, BYTE],         !Moduleflags, idlng, moduleid
        globlist : VECTOR [2],                                  !Listhead for globals to insert
        delist : VECTOR [2],                                    !Listhead for globals to delete
        compilecods : BBLOCK [5 * dsc$c_s_bln] INITIAL          !Name the compilation completion codes
                            (STRINGDESC ('success'),
                             STRINGDESC ('warnings'),
                             STRINGDESC ('errors'),
                                        STRINGDESC ('fatal errors'),
                                        STRINGDESC ('illegal compilation code'));

BIND
    modnamlng = mod_name [0] : BYTE,                                !Name the module name length
    modulename = mod_name [1] : VECTOR [,BYTE],                     ! and the module name
    moduleflags = moduledata [0] : BYTE,                            !Name module flags byte
    idlng = moduledata [1] : BYTE,                                  !Length of module ident
    moduleid = moduledata [2] : VECTOR [,BYTE],                     !Name module ident
    reclng = recdesc [dsc$w_length] : WORD,                         !Name the length of the record
    objrec = recdesc [dsc$a_pointer] : REF BBLOCK,                  !and the pointer
    objvec = recdesc [dsc$a_pointer] : REF VECTOR [,BYTE],
    recdispatch = PLIT(                                             !Set up maximum allowed record type
                prohdr,                                             !0 - module header
                progsd,                                             !1 - gsd records
                protir,                                             !2 - tir
                proeom,                                             !3 - end of module
                prorec,                                             !4 - dbg - check sequence and copy
                prorec,                                             !5 - tbt - check sequence and copy
                prorec,                                             !6 - lnk - check sequence and copy
                proeom) : VECTOR;                                   !7 - eomw
BUILTIN
    INSQUE,
    REMQUE;
```

```
%SBTTL  'LIB-INPUT_OBJ';

GLOBAL ROUTINE lib_input_obj =
BEGIN
!
! Process an object file
!
LOCAL
    hdrblkcnt,
    symdsc : REF BBLOCK,
    status;

IF .lib$gl_ctlmsk [lib$v_shrstb]                    !If processing shareable image stb
THEN BEGIN
    lib$al_rab [rab$l_bkt] = 1;                     !Set to read block 1
    lib$al_rab [rab$w_usz] = 512;                   ! and only block 1
    rms_perform ($READ (RAB = lib$al_rab),          !Read the image header
            lib$_readerr,                           ! report any error
            .lib$al_rab [rab$l_stv], 1, lib$gl_inpfdb [fdb$l_namdesc]);

    IF .lib$al_rab [rab$w_rsz] NEQ 512              ! Image header is 512 bytes long
        OR (
            BIND
                header = .lib$al_rab [rab$l_ubf] : BBLOCK;

            IF .header[ihd$b_imgtype] NEQ ihd$k_lim     ! type must agree
                OR .header[ihd$w_majorid] NEQ ihd$k_majorid ! major header id must match
                OR .header[ihd$w_minorid] GTRU ihd$k_minorid   ! minor id must not be greater
                OR .header[ihd$w_size] GTRU MAXU((.header[ihd$w_patchoff]
                        + ihp$k_length),ihd$k_length+
                            iha$k_length+ihs$k_length+ihi$k_length) ! Header fixed part must be
                                                        ! and contained in header
                OR (hdrblkcnt = .header[ihd$b_hdrblkcnt]-1) LSS 0
                OR (symdsc = header + .header[ihd$w_symdbgoff]) ! GST descriptor must be
                        GEQU (header + .header[ihd$w_size])     ! contained in header
                OR (.symdsc[ihs$w_gstrecs]) LSSU 3              ! Must be at least 3 blocks
                OR (.symdsc[ihs$l_gstvbn]) LEQU                 ! and must be beyond header blocks
                                            (.hdrblkcnt + 2)
                    THEN true                                   !It's not a shareable image
                    ELSE (shrgsmatch = .header[ihd$l_ident];!It's a shareable image, so save the gsmatch
                            false))
        THEN BEGIN
            SIGNAL (lib$_notshrimg, 1, lib$gl_inpfdb [fdb$l_namdesc]);
            RETURN lib$_notshrimg;
            END;
    lib$al_rab [rab$b_rac] = rab$c_rfa;                         !Set to point to object file
    IF (lib$al_rab [rab$l_rfa0] = .symdsc [ihs$l_gstvbn]) NEQ 0 ! which is the symbol table
    THEN BEGIN
        lib$al_rab [rab$w_rfa4] = 0;                            ! on a block boundary
        rms_perform ($FIND (RAB = lib$al_rab),
                    lib$_readerr, 1, lib$gl_inpfdb [fdb$l_namdesc]);
        lib$al_rab [rab$b_rac] = rab$c_seq;                     !Reset to sequentioal
        END
    ELSE BEGIN
        SIGNAL (lib$_nosymbols,1,lib$gl_inpfdb [fdb$l_namdesc]);
        RETURN true
        END;
```

```
;   275             1333  2         END;
;   276             1334  2 status = profile ();
;   277             1335  2 IF NOT .status                                   !Clean up if an error
;   278             1336  2     THEN finish_object (false);
;   279             1337  2 RETURN .status
;   280             1338  1 END;                                             !Of lib_input_obj


                                                                                       .TITLE  LIB_INPUTOBJ
                                                                                       .IDENT  \V04-000\

                                                                                       .PSECT  $PLIT$,NOWRT,NOEXE,2

                                   00   73   73   65   63   63   75   73   00000 P.AAA:  .ASCII  \success\<0>                                  ;
                                   73   67   6E   69   6E   72   61   77   00008 P.AAB:  .ASCII  \warnings\                                     ;
                                   00   00   73   72   6F   72   72   65   00010 P.AAC:  .ASCII  \errors\<0><0>                                 ;
                 73   72   6F   72   72   65   20   6C   61   74   61   66   00018 P.AAD:  .ASCII  \fatal errors\                                 ;
61   6C   69   70   6D   6F   63   20   6C   61   67   65   6C   6C   69   00024 P.AAE:  .ASCII  \illegal compilation code\                     ;
                              65   64   6F   63   20   6E   6F   69   74   00033                                                                ;
                                                              00000008   0003C         .LONG   8                                              ;
00000000V 00000000V 00000000V 00000000V 00000000V 00000000V 00040 P.AAF:  .ADDRESS PROHDR, PROGSD, PROTIR, PROEOM, PROREC, -         ;
                                                  00000000V 00000000V 00058                 PROREC, PROREC, PROEOM                      ;

                                                                                       .PSECT  $OWN$,NOEXE,2

                                                                         00000 SHRGSMATCH:
                                                                                       .BLKB   4
                                                                         00004 OPERATION:
                                                                                       .BLKB   4
                                                                         00008 MHDSEEN:.BLKB   4
                                                                         0000C LNMSEEN:.BLKB   4
                                                                         00010 DUPSEEN:.BLKB   4
                                                                         00014 GSDOFFSET:
                                                                                       .BLKB   4
                                                                         00018 SYMBOLSTRING:
                                                                                       .BLKB   4
                                                                         0001C RECDESC:.BLKB   8
                                                                         00024 LASTRECTYP:
                                                                                       .BLKB   4
                                                              00000003   00028 CURRECTYP:
                                                                                       .LONG   3                                      ;
                                                              00000800   0002C MAXRECLNG:
                                                                                       .LONG   2048                                   ;
                                                                         00030 MOD_NAME:
                                                                                       .BLKB   32
                                                                         00050 MODULERFA:
                                                                                       .BLKB   6
                                                                         00056         .BLKB   2
                                                                         00058 OLDMODRFA:
                                                                                       .BLKB   6
                                                                         0005E         .BLKB   2
                                                                         00060 REPLACING:
                                                                                       .BLKB   4
                                                              00000000   00064 MODULEDESC:
                                                                                       .LONG   0                                      ;
                                                              00000000'  00068         .ADDRESS MOD_NAME+1                            ;
                                                                         0006C MODULEDATA:
```

```
                                   .BLKB    33
                    0008D          .BLKB    3
                    00090 GLOBLIST:
                                   .BLKB    8
                    00098 DELIST:  .BLKB    8
          00000007  000A0 COMPILECODS:
                                   .LONG    7
          00000000' 000A4          .ADDRESS P.AAA
          00000008  000A8          .LONG    8
          00000000' 000AC          .ADDRESS P.AAB
          00000006  000B0          .LONG    6
          00000000' 000B4          .ADDRESS P.AAC
          0000000C  000B8          .LONG    12
          00000000' 000BC          .ADDRESS P.AAD
          00000018  000C0          .LONG    24
          00000000' 000C4          .ADDRESS P.AAE

                          MODNAMLNG=          MOD_NAME
                          MODULENAME=         MOD_NAME+1
                          MODULEFLAGS=        MODULEDATA
                          IDLNG=              MODULEDATA+1
                          MODULEID=           MODULEDATA+2
                          RECLNG=             RECDESC
                          OBJREC=             RECDESC+4
                          OBJVEC=             RECDESC+4
                          RECDISPATCH=        P.AAF
                                   .EXTRN   LBR$GL_RMSSTV, LIB$GL_OBJMODIX
                                   .EXTRN   LIB$GL_OBJGSDIX
                                   .EXTRN   LIB$GL_RECOUNT, LIB$AL_RAB
                                   .EXTRN   LIB$GL_TYPE, LIB$GL_KEYSIZE
                                   .EXTRN   LIB$GL_CTLMSK, LIB$GL_LIBFDB
                                   .EXTRN   LIB$GL_INPFDB, LIB$GL_LIBCTL
                                   .EXTRN   LIB_GET_MEM, LIB_GET_ZMEM
                                   .EXTRN   LIB_FREE_MEM, LIB_LOG_OP
                                   .EXTRN   LIB_LOG_OPD, LBR$SEARCH
                                   .EXTRN   LBR$DELETE_DATA
                                   .EXTRN   LBR$PUT_RECORD, LBR$PUT_END
                                   .EXTRN   LBR$LOOKUP_KEY, LBR$SET_INDEX
                                   .EXTRN   LBR$INSERT_KEY, LBR$SET_MODULE
                                   .EXTRN   LBR$REPLACE_KEY
                                   .EXTRN   LBR$DELETE_KEY, GET_RECORD
                                   .EXTRN   LIB$_NOTSHRIMG, LIB$_NOSYMBOLS
                                   .EXTRN   LIB$_RECLNG, LIB$_RECTYP
                                   .EXTRN   LIB$_NOEOM, LIB$_STRLVL
                                   .EXTRN   LIB$_MODNAMLNG, LIB$_INDEXERR
                                   .EXTRN   LIB$_INSERTED, LIB$_REPLACED
                                   .EXTRN   LIB$_DUPMODULE, LIB$_GSDTYP
                                   .EXTRN   LIB$_SPNAMLNG, LIB$_SYMNAMLNG
                                   .EXTRN   LIB$_DUPGLOBAL, LIB$_COMCOD
                                   .EXTRN   LIB$_MHDERR, LIB$_INSERTERR
                                   .EXTRN   LIB$_DELKEYERR, LIB$_DELDATERR
                                   .EXTRN   LIB$_SEQNCE, SYS$READ
                                   .EXTRN   SYS$FIND

                                   .PSECT   $CODE$,NOWRT,2

               00FC 00000          .ENTRY   LIB_INPUT_OBJ, Save R2,R3,R4,R5,R6,R7        ; 1278
```

```
                          57 00000000G   8F  D0 00002          MOVL    #LIB$_NOTSHRIMG, R7                  :
                          56     0000G   CF  9E 00009          MOVAB   LIB$GL_INPFDB, R6                    :
                          55 00000000G   00  9E 0000E          MOVAB   LIB$SIGNAL, R5                       :
                          54     0000G   CF  9E 00015          MOVAB   LIB$AL_RAB, R4                       :
            03     0000G  CF             05  E0 0001A          BBS     #5, LIB$GL_CTLMSK, 1$                : 1288
                                       00EA  31 00020          BRW     8$                                   :
                    38    A4             01  D0 00023 1$:      MOVL    #1, LIB$AL_RAB+56                    : 1290
                    20    A4      0200   8F  B0 00027          MOVW    #512, LIB$AL_RAB+32                  : 1291
                                         54  DD 0002D          PUSHL   R4                                   : 1294
              00000000G   00             01  FB 0002F          CALLS   #1, SYS$READ                         :
                          14             50  E8 00036          BLBS    STATUS, 2$                           :
                                  0C     A4  DD 00039          PUSHL   LIB$AL_RAB+12                        :
                                         50  DD 0003C          PUSHL   STATUS                               :
            7E            66             10  C1 0003E          ADDL3   #16, LIB$GL_INPFDB, -(SP)            :
                                         01  DD 00042          PUSHL   #1                                   :
                             008610B2    8F  DD 00044          PUSHL   #8786098                             :
                          65             05  FB 0004A          CALLS   #5, LIB$SIGNAL                       :
                  0200    8F      22     A4  B1 0004D 2$:      CMPW    LIB$AL_RAB+34, #512                  : 1296
                          66             12     00053          BNEQ    4$                                   :
                          51      24     A4  D0 00055          MOVL    LIB$AL_RAB+36, R1                    : 1299
                          02      11     A1  91 00059          CMPB    17(R1), #2                           : 1301
                          5C             12     0005D          BNEQ    4$                                   :
                  3230    8F      0C     A1  B1 0005F          CMPW    12(R1), #12848                       : 1302
                          54             12     00065          BNEQ    4$                                   :
                  3530    8F      0E     A1  B1 00067          CMPW    14(R1), #13616                       : 1303
                          4C             1A     0006D          BGTRU   4$                                   :
                          50      08     A1  3C 0006F          MOVZWL  8(R1), R0                            : 1304
                          50             2C  C0 00073          ADDL2   #44, R0                              :
              000000A8    8F             50  D1 00076          CMPL    R0, #168                             : 1306
                          04             1E     0007D          BGEQU   3$                                   :
                          50      A8     8F  9A 0007F          MOVZBL  #168, R0                             :
    50      61            10             00  ED 00083 3$:      CMPZV   #0, #16, (R1), R0                    : 1304
                          31             1A     00088          BGTRU   4$                                   :
                          50      10     A1  9A 0008A          MOVZBL  16(R1), HDRBLKCNT                    : 1308
                          50             D7     0008E          DECL    HDRBLKCNT                            :
                          29             19     00090          BLSS    4$                                   :
                          53      04     A1  3C 00092          MOVZWL  4(R1), SYMDSC                        : 1309
                          53             51  C0 00096          ADDL2   R1, SYMDSC                           :
                          52             61  3C 00099          MOVZWL  (R1), R2                             : 1310
                          52             51  C0 0009C          ADDL2   R1, R2                               :
                          52             53  D1 0009F          CMPL    SYMDSC, R2                           :
                          17             1E     000A2          BGEQU   4$                                   :
                          03      0A     A3  B1 000A4          CMPW    10(SYMDSC), #3                       : 1311
                          11             1F     000A8          BLSSU   4$                                   :
                          50             02  C0 000AA          ADDL2   #2, R0                               : 1313
                          50      04     A3  D1 000AD          CMPL    4(SYMDSC), R0                        :
                          08             1B     000B1          BLEQU   4$                                   :
                  0000'   CF      24     A1  D0 000B3          MOVL    36(R1), SHRGSMATCH                   : 1315
                          0F             11     000B9          BRB     5$                                   :
            7E            66             10  C1 000BB 4$:      ADDL3   #16, LIB$GL_INPFDB, -(SP)            : 1318
                                         01  DD 000BF          PUSHL   #1                                   :
                                         57  DD 000C1          PUSHL   R7                                   :
                          65             03  FB 000C3          CALLS   #3, LIB$SIGNAL                       :
                          50             57  D0 000C6          MOVL    R7, R0                               : 1319
                                             04 000C9          RET                                          :
                    1E    A4             02  90 000CA 5$:      MOVB    #2, LIB$AL_RAB+30                    : 1321
                    10    A4      04     A3  D0 000CE          MOVL    4(SYMDSC), LIB$AL_RAB+16             : 1322
```

```
                                        25  13 000D3          BEQL    7$                                        ;
                               14       A4  B4 000D5          CLRW    LIB$AL_RAB+20                             ; 1324
                                        54  DD 000D8          PUSHL   R4                                        ; 1326
                  00000000G  00         01  FB 000DA          CALLS   #1, SYS$FIND                              ;
                             11         50  E8 000E1          BLBS    STATUS, 6$                                ;
                                        01  DD 000E4          PUSHL   #1                                        ;
                                        50  DD 000E6          PUSHL   STATUS                                    ;
               7E            66         10  C1 000E8          ADDL3   #16, LIB$GL_INPFDB, -(SP)                 ;
                                008610B2 8F  DD 000EC         PUSHL   #8786098                                  ;
                             65         04  FB 000F2          CALLS   #4, LIB$SIGNAL                            ;
                               1E       A4  94 000F5 6$:      CLRB    LIB$AL_RAB+30                             ; 1327
                                        13  11 000F8          BRB     8$                                        ; 1322
               7E            66         10  C1 000FA 7$:      ADDL3   #16, LIB$GL_INPFDB, -(SP)                 ; 1330
                                        01  DD 000FE          PUSHL   #1                                        ;
                                00000000G 8F  DD 00100        PUSHL   #LIB$_NOSYMBOLS                           ;
                             65         03  FB 00106          CALLS   #3, LIB$SIGNAL                            ;
                             50         01  D0 00109          MOVL    #1, R0                                    ; 1331
                                            04 0010C          RET                                               ;
                     0000V   CF         00  FB 0010D 8$:      CALLS   #0, PROFILE                               ; 1334
                             52         50  D0 00112          MOVL    R0, STATUS                                ;
                             07         52  E8 00115          BLBS    STATUS, 9$                                ; 1335
                                        7E  D4 00118          CLRL    -(SP)                                     ; 1336
                     0000V   CF         01  FB 0011A          CALLS   #1, FINISH_OBJECT                         ;
                             50         52  D0 0011F 9$:      MOVL    STATUS, R0                                ; 1337
                                            04 00122          RET                                               ; 1338

; Routine Size:  291 bytes,     Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL  'profile';
1
1 ROUTINE profile =
2 BEGIN
2 !
2 ! Read and process all required object module records of the file just opened
2 ! that is, keep reading records to end of file.
2 !
2 !
2 !
2 LOCAL
2     status;
2
2 modnamlng = 0;                                      !Zero module name
2 modulerfa [rfa$l_vbn] = 0;                          !Clear VBN
2 mhdseen = false;
2 lnmseen = false;
2 currectyp = obj$c_eom;                              !Init record to end of module type
2 globlist [0] = globlist [0];                        !Init globals listhead
2 globlist [1] = globlist [0];
2 delist [0] = delist [0];
2 delist [1] = delist [0];
2 moduleflags = 0;                                    ! Zero module flags
2 WHILE (status = get_record (recdesc)) NEQ rms$_eof ! While there are more records
3 DO BEGIN
3         lib$gl_recount = .lib$gl_recount + 1;   ! Count the record
3         IF .reclng GTRU .maxreclng              ! And if its length is illegal
4         THEN BEGIN
4             SIGNAL (lib$_reclng, 3, .reclng,    !   then signal the error and give up on this file
4                     modnamlng, lib$gl_inpfdb [fdb$l_namdesc]);
4             RETURN lib$_reclng;
3             END;
3         lastrectyp = .currectyp;                ! Copy old current to last type
3         currectyp = .objrec [obj$b_rectyp];     ! And get new type
3         IF .currectyp LSSU .recdispatch [-1]    ! Check it is legal and if
3         THEN
4             BEGIN
4             !
4             !     If a duplicate module is being processed then ignore record
4             !     unless it is a new module header record.
4             !
5             IF (NOT .dupseen)
4             THEN
4                 perform ((.recdispatch [.currectyp]) ()); ! So dispatch to record specific routine
5             IF .dupseen AND (.currectyp EQL 3)
4             THEN
4                 dupseen = false;
4             END
3         ELSE
4             BEGIN
4             SIGNAL (lib$_rectyp, 3, .currectyp, !If unknown, signal and give up
4                             modnamlng, lib$gl_inpfdb [fdb$l_namdesc]);
4             RETURN lib$_rectyp;
3             END;
3         IF .lib$gl_ctlmsk [lib$v_shrstb]
3             AND .currectyp EQL obj$c_eom
3             THEN EXITLOOP;
```

```
                                                                        D 12
LIB_INPUTOBJ                                                            16-Sep-1984 01:57:57    VAX-11 Bliss-32 V4.0-742               Page  12
V04-000              profile                                            14-Sep-1984 12:38:04    [LIBRAR.SRC]INPUTOBJ.B32;1                    (4)

;  339          1396  2 END;                                   ! Of records loop
;  340          1397  2 IF .currectyp NEQ obj$c_eom                                   ! All done, did we end with eom?
;  341          1398  3 THEN BEGIN
;  342          1399  3     SIGNAL (lib$_noeom, 2, modnamlng, lib$gl_inpfdb [fdb$l_namdesc]); !no, signal and return
;  343          1400  3     RETURN lib$_noeom;
;  344          1401  2     END;
;  345          1402  2 RETURN true                                          ! Finally return after no more
;  346          1403  1 END;                                                 ! Of lib_input_obj



                                                 01FC 00000 PROFILE:.WORD    Save R2,R3,R4,R5,R6,R7,R8                   ; 1341
                        58        0000G   CF   9E 00002          MOVAB    LIB$GL_INPFDB, R8                           ;
                        57 00000000G      8F   D0 00007          MOVL     #LIB$_NOEOM, R7                             ;
                        56 00000000G      8F   D0 0000E          MOVL     #LIB$_RECTYP, R6                            ;
                        55 00000000G      8F   D0 00015          MOVL     #LIB$_RECLNG, R5                            ;
                        54 00000000G      00   9E 0001C          MOVAB    LIB$SIGNAL, R4                              ;
                        53        0000'   CF   9E 00023          MOVAB    CURRECTYP, R3                               ;
                                  08      A3   94 00028          CLRB     MODNAMLNG                                   ; 1352
                                  28      A3   D4 0002B          CLRL     MODULERFA                                   ; 1353
                                  E0      A3   7C 0002E          CLRQ     MHDSEEN                                     ; 1354
                        63                03   D0 00031          MOVL     #3, CURRECTYP                               ; 1356
                  68    A3        68      A3   9E 00034          MOVAB    GLOBLIST, GLOBLIST                          ; 1357
                  6C    A3        68      A3   9E 00039          MOVAB    GLOBLIST, GLOBLIST+4                        ; 1358
                  70    A3        70      A3   9E 0003E          MOVAB    DELIST, DELIST                              ; 1359
                  74    A3        70      A3   9E 00043          MOVAB    DELIST, DELIST+4                            ; 1360
                                  44      A3   94 00048          CLRB     MODULEFLAGS                                 ; 1361
                                  F4      A3   9F 0004B 1$:      PUSHAB   RECDESC                                     ; 1362
               0000G    CF                01   FB 0004E          CALLS    #1, GET_RECORD                              ;
                        52                50   D0 00053          MOVL     R0, STATUS                                  ;
           0001827A     8F                52   D1 00056          CMPL     STATUS, #98938                              ;
                                          75   13 0005D          BEQL     8$                                          ;
                                  0000G   CF   D6 0005F          INCL     LIB$GL_RECOUNT                              ; 1364
  04    A3      F4    A3          10      00   ED 00063          CMPZV    #0, #16, RECLNG, MAXRECLNG                  ; 1365
                                          16   1B 0006A          BLEQU    2$                                          ;
              7E                  68      10   C1 0006C          ADDL3    #16, LIB$GL_INPFDB, -(SP)                   ; 1368
                                  08      A3   9F 00070          PUSHAB   MODNAMLNG                                   ; 1367
                        7E        F4      A3   3C 00073          MOVZWL   RECLNG, -(SP)                               ; 1368
                                          03   DD 00077          PUSHL    #3                                          ;
                                          55   DD 00079          PUSHL    R5                                          ;
                        64                05   FB 0007B          CALLS    #5, LIB$SIGNAL                              ;
                        50                55   D0 0007E          MOVL     R5, R0                                      ; 1369
                                               04 00081          RET                                                  ;
                  FC    A3                63   D0 00082 2$:      MOVL     CURRECTYP, LASTRECTYP                       ; 1371
                        63        F8      B3   9A 00086          MOVZBL   @OBJREC, CURRECTYP                          ; 1372
                        50                63   D0 0008A          MOVL     CURRECTYP, R0                               ; 1373
               0000'    CF                50   D1 0008D          CMPL     R0, RECDISPATCH-4                           ;
                        1E                1E   1E 00092          BGEQU    4$                                          ;
                        10        E8      A3   E8 00094          BLBS     DUPSEEN, 3$                                 ; 1380
                        50        0000'CF 40   D0 00098          MOVL     RECDISPATCH[R0], R0                         ; 1382
                        60                00   FB 0009E          CALLS    #0, (R0)                                    ;
                        4A                50   E9 000A1          BLBC     STATUS, 10$                                 ;
                        1E        E8      A3   E9 000A4          BLBC     DUPSEEN, 5$                                 ; 1383
                        03                63   D1 000A8 3$:      CMPL     CURRECTYP, #3                               ;
                                          19   12 000AB          BNEQ     5$                                          ;
```

```
                                     E8   A3  D4 000AD          CLRL    DUPSEEN                          ; 1385
                                          14  11 000B0          BRB     5$                               ; 1373
               7E          68             10  C1 000B2 4$:      ADDL3   #16, LIB$GL_INPFDB, -(SP)        ; 1390
                                     08   A3  9F 000B6          PUSHAB  MODNAMLNG                        ; 1389
                                          50  DD 000B9          PUSHL   R0                               ; 1390
                                          03  DD 000BB          PUSHL   #3                               ;
                                          56  DD 000BD          PUSHL   R6                               ;
                           64             05  FB 000BF          CALLS   #5, LIB$SIGNAL                   ;
                           50             56  D0 000C2          MOVL    R6, R0                           ; 1391
                                          04  000C5             RET                                      ;
               03   0000G  CF             05  E0 000C6 5$:      BBS     #5, LIB$GL_CTLMSK, 7$            ; 1393
                                        FF7C  31 000CC 6$:      BRW     1$                               ;
                           03             63  D1 000CF 7$:      CMPL    CURRECTYP, #3                    ; 1394
                                          F8  12 000D2          BNEQ    6$                               ;
                           03             63  D1 000D4 8$:      CMPL    CURRECTYP, #3                    ; 1397
                                          12  13 000D7          BEQL    9$                               ;
               7E          68             10  C1 000D9          ADDL3   #16, LIB$GL_INPFDB, -(SP)        ; 1399
                                     08   A3  9F 000DD          PUSHAB  MODNAMLNG                        ;
                                          02  DD 000E0          PUSHL   #2                               ;
                                          57  DD 000E2          PUSHL   R7                               ;
                           64             04  FB 000E4          CALLS   #4, LIB$SIGNAL                   ;
                           50             57  D0 000E7          MOVL    R7, R0                           ; 1400
                                          04  000EA             RET                                      ;
                           50             01  D0 000EB 9$:      MOVL    #1, R0                           ; 1402
                                          04  000EE 10$:        RET                                      ; 1403

; Routine Size:  239 bytes,    Routine Base:  $CODE$ + 0123
```

```
1 %SBTTL  'prohdr';
1
1 ROUTINE prohdr =
2 BEGIN
2 !
2 !++
2 !       process module header records as follows:
2 !               (1) validate sequence
2 !               (2) ignore all but main module headers
2 !               (3) verify structure level is less than
2 !                   or equal to obj$c_strlvl
2 !               (4) verify maximum record length
2 !                   parameter is less than or equal to
2 !                   obj$c_maxrecsiz
2 !               (5) record maximum record length parameter
2 !                   for checking subsequent records
2 !               (6) check module title > 0 and less than or
2 !                   equal to sym$c_maxlng characters
2 !               (7) copy the module title
2 !--
2 !
2 LOCAL
2     txtrfa : BBLOCK [rfa$c_length];
2
2 BIND
2       modidstring = objrec [mhd$t_name] + .objrec [mhd$b_namlng] : VECTOR [,BYTE];
2
2 perform (segchk ());
2 IF .objrec [obj$b_subtyp] NEQ obj$c_hdr_mhd                    !Ignore all headers except main header
2 THEN IF NOT .lib$gl_ctlmsk [lib$v_shrstb]
2       THEN RETURN copyrec ()                                   !Just copy them
2       ELSE RETURN true;
2
2 IF .objrec [mhd$b_strlvl] GTRU obj$c_strlvl                    ! Compare its obj format
3 THEN BEGIN
3     SIGNAL (lib$_strlvl, 3, .objrec [mhd$b_strlvl], modnamlng,
3             lib$gl_inpfdb [fdb$l_namdesc]);
3     RETURN lib$_strlvl;
2     END;
2 IF (maxreclng = .objrec [mhd$w_recsiz]) GTRU obj$c_maxrecsiz   ! Compare max with max allowed
3 THEN BEGIN
3     SIGNAL (lib$_reclng, 3, .maxreclng, modnamlng,
3             lib$gl_inpfdb [fdb$l_namdesc]);
3     RETURN lib$_reclng;
2     END;
2 IF .objrec [mhd$b_namlng] GTRU .lib$gl_keysize                 ! Check module name is within legal
2 OR .objrec [mhd$b_namlng] EQL 0                                ! Length range
3 THEN BEGIN
3     SIGNAL (lib$_modnamlng, 3, objrec [mhd$b_namlng], .objrec [mhd$b_namlng],
3             lib$gl_inpfdb [fdb$l_namdesc]);
3     RETURN lib$_modnamlng;
2     END;
2 modnamlng = .objrec [mhd$b_namlng];            !Copy length of module name
2 CH$MOVE  (.objrec [mhd$b_namlng], objrec [mhd$t_name], modulename);
2 IF .lib$gl_ctlmsk [lib$v_shrstb]
3 THEN BEGIN
3     idlng = 4;                                 !GSMATCH is 4 bytes long
```

```
;   405          1461  3        CH$MOVE(4,shrgsmatch,moduleid);              !Copy the GSMATCH into module header data
;   406          1462  3        END
;   407          1463  3    ELSE BEGIN
;   408          1464  3        idlng = MINU (sym$c_maxlng, .modidstring [0]);
;   409          1465  3        CH$MOVE (.modidstring [0], modidstring [1], moduleid);
;   410          1466  2        END;
;   411          1467  2    moduledesc [dsc$w_length] = .modnamlng;
;   412        P 1468  2    perform (lbr$set_index (lib$gl_libctl, lib$gl_objmodix),
;   413          1469  2                lib$_indexerr, 1, lib$gl_libfdb [fdb$l_namdesc]);
;   414          1470  2    replacing = false;
;   415          1471  2    operation = lib$_inserted;
;   416          1472  2
;   417          1473  2    CH$FILL (0, rfa$c_length, oldmodrfa);                          ! initialize rfa
;   418          1474  2    IF lbr$lookup_key (lib$gl_libctl, moduledesc, oldmodrfa)       !If in library already
;   419          1475  2    THEN IF .lib$gl_ctlmsk [lib$v_replace]                         !If replace
;   420          1476  2    !
;   421          1477  2    ! Key in index, and replacing.  Find globals that belong with old
;   422          1478  2    ! module and put on list.
;   423          1479  2    !
;   424          1480  3        THEN BEGIN
;   425          1481  3            lbr$search (lib$gl_libctl, lib$gl_objgsdix, oldmodrfa, delsym);
;   426          1482  3            replacing = true;
;   427          1483  3            operation = lib$_replaced;                             !Set for proeom
;   428          1484  3            END
;   429          1485  3        ELSE BEGIN
;   430          1486  3            SIGNAL (lib$_dupmodule, 3, modnamlng, lib$gl_inpfdb [fdb$l_namdesc],
;   431          1487  3                        lib$gl_libfdb [fdb$l_namdesc]);
;   432          1488  3            dupseen = true;
;   433          1489  3            RETURN true;
;   434          1490  2            END;
;   435          1491  2
;   436          1492  2    perform (copyrec ());                                          !Copy record to library
;   437          1493  2
;   438          1494  2    RETURN true
;   439          1495  1    END;                     ! OF prohdr





                                           0FFC 00000 PROHDR: .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 1406
                        5B 00000000G   8F   D0 00002          MOVL    #LIB$_RECLNG, R11                           ;
                        5A 00000000G   8F   D0 00009          MOVL    #LIB$_STRLVL, R10                           ;
                        59     0000G   CF   9E 00010          MOVAB   LIB$GL_INPFDB, R9                           ;
                        58 00000000G   00   9E 00015          MOVAB   LIB$SIGNAL, R8                              ;
                        57     0000'   CF   9E 0001C          MOVAB   OBJREC, R7                                  ;
                        5E             08   C2 00021          SUBL2   #8, SP                                      ;
                        51             67   D0 00024          MOVL    OBJREC, R1                                  ; 1429
                        50        05   A1   9A 00027          MOVZBL  5(R1), R0                                   ;
                        56        06 A140   9E 0002B          MOVAB   6(R1)[R0], R6                               ;
                 0000V  CF             00   FB 00030          CALLS   #0, SEQCHK                                  ; 1431
                        01             50   E8 00035          BLBS    STATUS, 1$                                  ;
                                            04 00038          RET                                                 ;
                        50             67   D0 00039 1$:      MOVL    OBJREC, R0                                  ; 1432
                                  01   A0   95 0003C          TSTB    1(R0)                                       ;
                                       0F   13 0003F          BEQL    3$                                          ;
          03     0000G  CF             05   E1 00041          BBC     #5, LIB$GL_CTLMSK, 2$                       ; 1433
```

```
                                                                          H 12
LIB_INPUTOBJ                                                    16-Sep-1984 01:57:57    VAX-11 Bliss-32 V4.0-742            Page  16
V04-000          prohdr                                         14-Sep-1984 12:38:04    [LIBRAR.SRC]INPUTOBJ.B32;1                (5)

                                                             0156 31 00047         BRW      14$                              ;
                                   0000V    CF                 00 FB 0004A 2$:     CALLS    #0, COPYREC                      ; 1434
                                                                  04 0004F         RET                                       ; 1435
                                            50                 67 D0 00050 3$:     MOVL     OBJREC, R0                       ; 1437
                                                        02     A0 95 00053         TSTB     2(R0)                            ;
                                                               16 13 00056         BEQL     4$                               ;
                            7E              69                 10 C1 00058         ADDL3    #16, LIB$GL_INPFDB, -(SP)        ; 1440
                                                        10     A7 9F 0005C         PUSHAB   MODNAMLNG                        ; 1439
                            7E                          02     A0 9A 0005F         MOVZBL   2(R0), -(SP)                     ; 1440
                                                               03 DD 00063         PUSHL    #3                               ;
                                                               5A DD 00065         PUSHL    R10                              ;
                                            68                 05 FB 00067         CALLS    #5, LIB$SIGNAL                   ;
                                            50                 5A D0 0006A         MOVL     R10, R0                          ; 1441
                                                                  04 0006D         RET                                       ;
                                            50                 67 D0 0006E 4$:     MOVL     OBJREC, R0                       ; 1443
                                            50          03     A0 3C 00071         MOVZWL   3(R0), R0                        ;
                                      0C    A7                 50 D0 00075         MOVL     R0, MAXRECLNG                    ;
                                    0800    8F                 50 B1 00079         CMPW     R0, #2048                        ;
                                                               15 1B 0007E         BLEQU    5$                               ;
                            7E              69                 10 C1 00080         ADDL3    #16, LIB$GL_INPFDB, -(SP)        ; 1446
                                                        10     A7 9F 00084         PUSHAB   MODNAMLNG                        ; 1445
                                                        0C     A7 DD 00087         PUSHL    MAXRECLNG                        ; 1446
                                                               03 DD 0008A         PUSHL    #3                               ;
                                                               5B DD 0008C         PUSHL    R11                              ;
                                            68                 05 FB 0008E         CALLS    #5, LIB$SIGNAL                   ;
                                            50                 5B D0 00091         MOVL     R11, R0                          ; 1447
                                                                  04 00094         RET                                       ;
                                            50                 67 D0 00095 5$:     MOVL     OBJREC, R0                       ; 1449
     0000G    CF          05    A0          08                 00 ED 00098         CMPZV    #0, #8, 5(R0), LIB$GL_KEYSIZE    ;
                                                               05 1A 000A0         BGTRU    6$                               ;
                                                        05     A0 95 000A2         TSTB     5(R0)                            ; 1450
                                                               1E 12 000A5         BNEQ     7$                               ;
                            7E              69                 10 C1 000A7 6$:     ADDL3    #16, LIB$GL_INPFDB, -(SP)        ; 1453
                                            7E          05     A0 9A 000AB         MOVZBL   5(R0), -(SP)                     ;
                                                        05     A0 9F 000AF         PUSHAB   5(R0)                            ; 1452
                                                               03 DD 000B2         PUSHL    #3                               ; 1453
                                                 00000000G     8F DD 000B4         PUSHL    #LIB$_MODNAMLNG                  ;
                                            68                 05 FB 000BA         CALLS    #5, LIB$SIGNAL                   ;
                                            50   00000000G     8F D0 000BD         MOVL     #LIB$_MODNAMLNG, R0              ; 1454
                                                                  04 000C4         RET                                       ;
                                            50                 67 D0 000C5 7$:     MOVL     OBJREC, R0                       ; 1456
                                      10    A7          05     A0 90 000C8         MOVB     5(R0), MODNAMLNG                 ;
                                            51          05     A0 9A 000CD         MOVZBL   5(R0), R1                        ; 1457
                      11    A7        06    A0                 51 28 000D1         MOVC3    R1, 6(R0), MODULENAME            ;
                            0B     0000G    CF                 05 E1 000D7         BBC      #5, LIB$GL_CTLMSK, 8$            ; 1458
                                      4D    A7                 04 90 000DD         MOVB     #4, IDLNG                        ; 1460
                                      4E    A7          E0     A7 D0 000E1         MOVL     SHRG$MATCH, MODULEID             ; 1461
                                                               18 11 000E6         BRB      10$                              ; 1458
                                            50                 66 9A 000E8 8$:     MOVZBL   (R6), R0                         ; 1464
                                            1F                 50 91 000EB         CMPB     R0, #31                          ;
                                                               03 1B 000EE         BLEQU    9$                               ;
                                            50                 1F D0 000F0         MOVL     #31, R0                          ;
                                      4D    A7                 50 90 000F3 9$:     MOVB     R0, IDLNG                        ;
                                            50                 66 9A 000F7         MOVZBL   (R6), R0                         ; 1465
                      4E    A7        01    A6                 50 28 000FA         MOVC3    R0, 1(R6), MODULEID              ;
                                      44    A7          10     A7 9B 00100 10$:    MOVZBW   MODNAMLNG, MODULEDESC            ; 1467
                                                     0000G     CF 9F 00105         PUSHAB   LIB$GL_OBJMODIX                  ; 1469
```

```
                                                                      I 12
LIB_INPUTOBJ                                                   16-Sep-1984 01:57:57    VAX-11 Bliss-32 V4.0-742              Page  17
V04-000          prohdr                                        14-Sep-1984 12:38:04    [LIBRAR.SRC]INPUTOBJ.B32;1                 (5)

                                           0000G  CF  9F 00109          PUSHAB  LIB$GL_LIBCTL                              ;
                         00000000G  00     02  FB 0010D          CALLS   #2, LBR$SET_INDEX                          ;
                                    13     50  E8 00114          BLBS    STATUS, 11$                                ;
                                           50  DD 00117          PUSHL   STATUS                                     ;
                  7E         0000G  CF     10  C1 00119          ADDL3   #16, LIB$GL_LIBFDB, -(SP)                  ;
                                           01  DD 0011F          PUSHL   #1                                         ;
                                00000000G  8F  DD 00121          PUSHL   #LIB$_INDEXERR                             ;
                                    68     04  FB 00127          CALLS   #4, LIB$SIGNAL                             ;
                                       40  A7  D4 0012A  11$:    CLRL    REPLACING                                  ; 1470
                              E4  A7 00000000G 8F  D0 0012D      MOVL    #LIB$_INSERTED, OPERATION                  ; 1471
         06               00        6E     00  2C 00135          MOVC5   #0, (SP), #0, #6, OLDMODRFA                ; 1473
                                       38  A7     0013A                                                             ;
                                       38  A7  9F 0013C          PUSHAB  OLDMODRFA                                  ; 1474
                                       44  A7  9F 0013F          PUSHAB  MODULEDESC                                 ;
                                     0000G CF  9F 00142          PUSHAB  LIB$GL_LIBCTL                              ;
                         00000000G  00     03  FB 00146          CALLS   #3, LBR$LOOKUP_KEY                         ;
                                    48     50  E9 0014D          BLBC    R0, 13$                                    ;
                  24         0000G  CF     05  E1 00150          BBC     #5, LIB$GL_CTLMSK+1, 12$                   ; 1475
                                     0000V CF  9F 00156          PUSHAB  DELSYM                                     ; 1481
                                       38  A7  9F 0015A          PUSHAB  OLDMODRFA                                  ;
                                     0000G CF  9F 0015D          PUSHAB  LIB$GL_OBJGSDIX                            ;
                                     0000G CF  9F 00161          PUSHAB  LIB$GL_LIBCTL                              ;
                         00000000G  00     04  FB 00165          CALLS   #4, LBR$SEARCH                             ;
                               40   A7     01  D0 0016C          MOVL    #1, REPLACING                              ; 1482
                               E4   A7 00000000G 8F D0 00170     MOVL    #LIB$_REPLACED, OPERATION                  ; 1483
                                           1E  11 00178          BRB     13$                                        ; 1475
                  7E         0000G  CF     10  C1 0017A  12$:    ADDL3   #16, LIB$GL_LIBFDB, -(SP)                  ; 1487
                  7E                69     10  C1 00180          ADDL3   #16, LIB$GL_INPFDB, -(SP)                  ; 1486
                                       10  A7  9F 00184          PUSHAB  MODNAMLNG                                  ;
                                           03  DD 00187          PUSHL   #3                                         ; 1487
                                00000000G  8F  DD 00189          PUSHL   #LIB$_DUPMODULE                            ;
                                    68     05  FB 0018F          CALLS   #5, LIB$SIGNAL                             ;
                               F0   A7     01  D0 00192          MOVL    #1, DUPSEEN                                ; 1488
                                           08  11 00196          BRB     14$                                        ; 1489
                             0000V  CF     00  FB 00198  13$:    CALLS   #0, COPYREC                                ; 1492
                                    03     50  E9 0019D          BLBC    STATUS, 15$                                ;
                                    50     01  D0 001A0  14$:    MOVL    #1, R0                                     ; 1494
                                               04 001A3  15$:    RET                                                ; 1495

; Routine Size:  420 bytes,    Routine Base:  $CODE$ + 0212
;   441         1496   1 %SBTTL  'delsym';
;   442         1497   1
;   443         1498   1 ROUTINE delsym (keydesc) =
;   444         1499   2 BEGIN
;   445         1500   2 !
;   446         1501   2 ! This routine is called by LBR$SEARCH for all globals that are in the module
;   447         1502   2 ! about to be replaced.  The names will be put on delist which will be scanned
;   448         1503   2 ! by prosymbol.
;   449         1504   2 !
;   450         1505   2 MAP
;   451         1506   2     keydesc : REF BBLOCK;
;   452         1507   2
;   453         1508   2 LOCAL
;   454         1509   2     keynb : REF BBLOCK;
;   455         1510   2
;   456         1511   2 perform (lib_get_mem (lnb$c_fixedsize + .keydesc [dsc$w_length], keynb));
;   457         1512   2 keynb [lnb$b_namlng] = .keydesc [dsc$w_length];
;   458         1513   2 keynb [lnb$b_flags] = 0;
;   459         1514   2 CH$MOVE (.keydesc [dsc$w_length], .keydesc [dsc$a_pointer], keynb [lnb$t_name]);
;   460         1515   2 INSQUE (.keynb, .delist [1]);
;   461         1516   2 RETURN true
;   462         1517   1 END;                                                    !Of delsym



                                        007C 00000 DELSYM: .WORD    Save R2,R3,R4,R5,R6                              ; 1498
                        5E          04    C2 00002         SUBL2    #4, SP                                           ;
                                    5E    DD 00005         PUSHL    SP                                               ; 1511
                        52      04  AC    D0 00007         MOVL     KEYDESC, R2                                      ;
                        7E          62    3C 0000B         MOVZWL   (R2), -(SP)                                      ;
                        6E          0A    C0 0000E         ADDL2    #10, (SP)                                        ;
                  0000G CF          02    FB 00011         CALLS    #2, LIB_GET_MEM                                  ;
                        18          50    E9 00016         BLBC     STATUS, 1$                                       ;
                        56          6E    D0 00019         MOVL     KEYNB, R6                                        ; 1512
                     09 A6          62    90 0001C         MOVB     (R2), 9(R6)                                      ;
                                08  A6    94 00020         CLRB     8(R6)                                            ; 1513
          0A   A6    04 B2          62    28 00023         MOVC3    (R2), @4(R2), 10(R6)                             ; 1514
                  0000' DF          66    0E 00029         INSQUE   (R6), @DELIST+4                                  ; 1515
                        50          01    D0 0002E         MOVL     #1, R0                                           ; 1516
                                          04 00031 1$:     RET                                                       ; 1517

; Routine Size:  50 bytes,    Routine Base:  $CODE$ + 03B6
;  464        1518  1 %SBTTL  'protir';
;  465        1519  1
;  466        1520  1 ROUTINE protir =
;  467        1521  2 BEGIN
;  468        1522  2 !
;  469        1523  2 ! This routine processes TIR records.  The OBJTIR flag is set in
;  470        1524  2 ! the module flags byte and the record is copied.
;  471        1525  2 !
;  472        1526  2 moduleflags = mhd$m_objtir;
;  473        1527  2 RETURN prorec ()
;  474        1528  1 END;                          ! Of protir



                                           0000 00000 PROTIR: .WORD    Save nothing                          ; 1520
                        0000'  CF       02   90 00002         MOVB     #2, MODULEFLAGS                       ; 1526
                        0000V  CF       00   FB 00007         CALLS    #0, PROREC                            ; 1527
                                             04 0000C         RET                                            ; 1528

; Routine Size:  13 bytes,    Routine Base:  $CODE$ + 03E8
```

```
1 %SBTTL  'progsd';
1
1 ROUTINE progsd =
2 BEGIN
2 !
2 !++
2 !
2 !       Verify GSD records and dispatch on the sub-types:
2 !               (0)  P-SECTION definition
2 !               (1)  Symbol definition/reference
2 !               (2)  Entry point definition
2 !               (3)  Procedure declaration
2 !               (4)  Symbol definition with word psect
2 !               (5)  Entry point definiton with word psect
2 !               (6)  Procedure definition with word psect
2 !               (7)  Random entity check
2 !               (8)  Environment definition
2 !               (9)  Local symbol definition/reference
2 !               (10) Local symbol entry point definition
2 !               (11) Local symbol procedure definition
2 !               (12) Shareable image psect definition
2 !
2 !--
2 !
2 BIND
2       gsddispatch = PLIT (            ! index         structure name
2                       propsectdef,    ! gsd_psc       gps$
2                       symbols,        ! gsd_sym       gsy$, srf$, sdf$
2                       entpnts,        ! gsd_epm
2                       procedef,       ! gsd_pro       pro$, fml$, arg$
2                       symbols,        ! gsd_symw      sdfw$
2                       pro_epmw,       ! gsd_epmw
2                       procedef,       ! gsd_prow
2                       pro_idc,        ! gsd_idc
2                       pro_env,        ! gsd_env
2                       pro_lsy,        ! gsd_lsy
2                       pro_lepm,       ! gsd_lepm
2                       pro_lpro,       ! gsd_lpro
2                       pro_spsc        ! gsd_spsc      sgps$
2                       ) : VECTOR;
2 !
2 LOCAL
2       gsdtype;
2 !
2 perform  (seqchk ());
2 gsdoffset = obj$c_subtyp;
2 !
2 WHILE .gsdoffset LSSU .reclng DO
3 BEGIN
3     IF ( gsdtype = .objvec [.gsdoffset]) GEQU .gsddispatch [-1]
4     THEN BEGIN
4         SIGNAL (lib$_gsdtyp, 3, modnamlng,
4                 lib$gl_inpfdb [fdb$l_namdesc], .gsdtype);
4         RETURN lib$_gsdtyp;
4         END
3     ELSE
3         perform (( .gsddispatch [.gsdtype]) ());
2     END;
```

```
;  533          1586  2
;  534          1587  2  IF NOT .lib$gl_ctlmsk [lib$v_shrstb]
;  535          1588  2     THEN RETURN copyrec ()
;  536          1589  2     ELSE RETURN true;
;  537          1590  2
;  538          1591  1  END;                 ! Of progsd


                                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                         0000000D  00060            .LONG   13                                                        ;
00000000V 00000000V 00000000V 00000000V 00000000V 00000000V  00064  P.AAG:   .ADDRESS PROPSECTDEF, SYMBOLS, ENTPNTS, PROCEDEF, -            ;
00000000V 00000000V 00000000V 00000000V 00000000V 00000000V  0007C                    SYMBOLS, PRO_EPMW, PROCEDEF, PRO_IDC, -           ;
                                                         00000000V 00094                    PRO_ENV, PRO_LSY, PRO_LEPM, PRO_[PRO, -         ;
                                                                                    PRO_SPSC                                          ;

                                                                  GSDDISPATCH=          P.AAG


                                                                            .PSECT  $CODE$,NOWRT,2

                                                      001C  00000  PROGSD:  .WORD   Save R2,R3,R4                                    ; 1531
                              54  00000000G    8F  D0  00002            MOVL    #LIB$_GSDTYP, R4                                 ;
                              53       0000'   CF  9E  00009            MOVAB   GSDOFFSET, R3                                    ;
                      0000V   CF               00  FB  0000E            CALLS   #0, SEQCHK                                       ; 1572
                              50               50  E9  00013            BLBC    STATUS, 5$                                       ;
                              63               01  D0  00016            MOVL    #1, GSDOFFSET                                    ; 1573
          63      08     A3   10               00  ED  00019  1$:      CMPZV   #0, #16, RECLNG, GSDOFFSET                       ; 1575
                              36               1B  0001F            BLEQU   3$                                               ;
                 50      0C   A3               63  C1  00021            ADDL3   GSDOFFSET, OBJVEC, R0                            ; 1577
                              52               60  9A  00026            MOVZBL  (R0), GSDTYPE                                    ;
                      0000'   CF               52  D1  00029            CMPL    GSDTYPE, GSDDISPATCH-4                           ;
                                               1A  1F  0002E            BLSSU   2$                                               ;
                                               52  DD  00030            PUSHL   GSDTYPE                                          ; 1580
                 7E   0000G   CF               10  C1  00032            ADDL3   #16, LIB$GL_INPFDB, -(SP)                        ;
                                          1C   A3  9F  00038            PUSHAB  MODNAMLNG                                        ; 1579
                                               03  DD  0003B            PUSHL   #3                                               ; 1580
                                               54  DD  0003D            PUSHL   R4                                               ;
                  00000000G   00               05  FB  0003F            CALLS   #5, LIB$SIGNAL                                   ;
                              50               54  D0  00046            MOVL    R4, R0                                           ; 1581
                                                   04  00049            RET                                                      ;
                              50       0000'CF42       D0  0004A  2$:      MOVL    GSDDISPATCH[GSDTYPE], R0                         ; 1584
                              60               00  FB  00050            CALLS   #0, (R0)                                         ;
                              C3               50  E8  00053            BLBS    STATUS, 1$                                       ;
                                                   04  00056            RET                                                      ;
                 06   0000G   CF               05  E0  00057  3$:      BBS     #5, LIB$GL_CTLMSK, 4$                            ; 1587
                      0000V   CF               00  FB  0005D            CALLS   #0, COPYREC                                      ; 1588
                                                   04  00062            RET                                                      ; 1589
                              50               01  D0  00063  4$:      MOVL    #1, R0                                           ;
                                                   04  00066  5$:      RET                                                      ; 1591

; Routine Size:  103 bytes,    Routine Base:  $CODE$ + 03F5
;  540       1592  1 %SBTTL  'propsectdef';
;  541       1593  1
;  542       1594  1 ROUTINE propsectdef =
;  543       1595  2 BEGIN
;  544       1596  2 !
;  545       1597  2 !++
;  546       1598  2 !       process P-section definitions as follows:
;  547       1599  2 !               (0) Check legal p-section name and alignment parameter
;  548       1600  2 !--
;  549       1601  2 !
;  550       1602  2 BIND
;  551       1603  2         psctdef = objvec [.gsdoffset] : BBLOCK;
;  552       1604  2 LOCAL
;  553       1605  2         length;
;  554       1606  2 !
;  555       1607  2 !       First check for legal P-section name and alignment
;  556       1608  2 !
;  557       1609  2 IF .psctdef [gps$b_namlng] GTRU sym$c_maxlng                    ! Check name within the legal
;  558       1610  2 OR .psctdef [gps$b_namlng] EQL 0                               ! Range for symbol and P-section
;  559       1611  3 THEN BEGIN
;  560       1612  3     SIGNAL (lib$_spnamlng, 3, modnamlng, lib$gl_inpfdb [fdb$l_namdesc],
;  561       1613  3             .psctdef [gps$b_namlng]);
;  562       1614  3     RETURN lib$_spnamlng;
;  563       1615  2     END;
;  564       1616  2 length = $BYTEOFFSET(gps$t_name) - $BYTEOFFSET(gps$t_start) +   ! Compute the offset of next GSD
;  565       1617  2             .psctdef [gps$b_namlng];
;  566       1618  2 gsdoffset = .gsdoffset + .length;                               ! From length of this
;  567       1619  2 RETURN true
;  568       1620  1 END;            ! Of propsectdef


                                      001C 00000 PROPSECTDEF:
                                                        .WORD   Save R2,R3,R4                           ; 1594
                 54      0000'   CF   9E 00002          MOVAB   GSDOFFSET, R4                           ;
                 53 00000000G    8F   D0 00007          MOVL    #LIB$_SPNAMLNG, R3                      ;
  52        0C   A4              64   C1 0000E          ADDL3   GSDOFFSET, OBJVEC, R2                   ; 1603
                 1F        08    A2   91 00013          CMPB    8(R2), #31                              ; 1609
                                 05   1A 00017          BGTRU   1$                                      ;
                           08    A2   95 00019          TSTB    8(R2)                                   ; 1610
                                 1C   12 0001C          BNEQ    2$                                      ;
                 7E        08    A2   9A 0001E  1$:     MOVZBL  8(R2), -(SP)                            ; 1613
  7E     0000G   CF              10   C1 00022          ADDL3   #16, LIB$GL_INPFDB, -(SP)               ; 1612
                           1C    A4   9F 00028          PUSHAB  MODNAMLNG                               ;
                                 03   DD 0002B          PUSHL   #3                                      ;
                                 53   DD 0002D          PUSHL   R3                                      ;
     00000000G   00              05   FB 0002F          CALLS   #5, LIB$SIGNAL                          ;
                 50              53   D0 00036          MOVL    R3, R0                                  ; 1614
                                      04 00039          RET                                             ;
                 50        08    A2   9A 0003A  2$:     MOVZBL  8(R2), LENGTH                           ; 1616
                 50              09   C0 0003E          ADDL2   #9, LENGTH                              ;
                 64              50   C0 00041          ADDL2   LENGTH, GSDOFFSET                       ; 1618
                 50              01   D0 00044          MOVL    #1, R0                                  ; 1619
                                      04 00047          RET                                             ; 1620
```

```
; Routine Size:  72 bytes,    Routine Base:  $CODE$ + 045C
;   570         1621  1 %SBTTL  'symbols';
;   571         1622  1
;   572         1623  1 ROUTINE symbols =
;   573         1624  2 BEGIN
;   574         1625  2 !
;   575         1626  2 LOCAL
;   576         1627  2         length;
;   577         1628  2 BIND
;   578         1629  2         symbolrec = objvec [.gsdoffset] : BBLOCK;
;   579         1630  2
;   580         1631  2
;   581         1632  2 IF NOT .symbolrec [gsy$v_def]
;   582         1633  3 THEN BEGIN
;   583         1634  3         length = $BYTEOFFSET(srf$t_name) - $BYTEOFFSET(srf$t_start) +
;   584         1635  3                 .symbolrec [srf$b_namlng];
;   585         1636  3         symbolstring = symbolrec [srf$b_namlng];        ! Point to the symbol string
;   586         1637  3     END
;   587         1638  3
;   588         1639  2 ELSE
;   589         1640  3     BEGIN
;   590         1641  3     IF .objvec [.gsdoffset] EQL obj$c_gsd_symw          ! If word psect
;   591         1642  3     THEN
;   592         1643  4         BEGIN
;   593         1644  4         length = $BYTEOFFSET(sdfw$t_name) - $BYTEOFFSET(sdfw$t_start) +
;   594         1645  4                 .symbolrec [sdfw$b_namlng];
;   595         1646  4         symbolstring = symbolrec [sdfw$b_namlng];       ! Point to the symbol
;   596         1647  4         END
;   597         1648  3     ELSE
;   598         1649  4         BEGIN
;   599         1650  4         length = $BYTEOFFSET(sdf$t_name) - $BYTEOFFSET(sdf$t_start) +
;   600         1651  4                 .symbolrec [sdf$b_namlng];
;   601         1652  4         symbolstring = symbolrec [sdf$b_namlng];        ! Point to the symbol
;   602         1653  3         END;
;   603         1654  3     IF NOT .symbolrec [gsy$v_weak]
;   604         1655  3     THEN
;   605         1656  3         perform (prosymbol ());
;   606         1657  2     END;
;   607         1658  2 gsdoffset = .gsdoffset + .length;       ! Update the gsd offset for next
;   608         1659  2 RETURN true
;   609         1660  1 END;                                    !Of symbols



                                        000C 00000 SYMBOLS:.WORD   Save R2,R3                              ; 1623
                        53      0000'   CF   9E 00002          MOVAB   SYMBOLSTRING, R3                        ;
        50        08    A3        FC    A3   C1 00007          ADDL3   GSDOFFSET, OBJVEC, R0                   ; 1629
        0D        02    A0              01   E0 0000D          BBS     #1, 2(R0), 1$                           ; 1632
                        52        04    A0   9A 00012          MOVZBL  4(R0), LENGTH                           ; 1634
                        52              05   C0 00016          ADDL2   #5, LENGTH                              ;
                        63        04    A0   9E 00019          MOVAB   4(R0), SYMBOLSTRING                     ; 1636
                                        29   11 0001D          BRB     4$                                      ; 1632
                  04                    60   91 0001F 1$:      CMPB    (R0), #4                                ; 1641
                                        0D   12 00022          BNEQ    2$                                      ;
                        52        0A    A0   9A 00024          MOVZBL  10(R0), LENGTH                          ; 1644
                        52              0B   C0 00028          ADDL2   #11, LENGTH                             ;
```

```
                                63       0A   A0  9E 0002B           MOVAB   10(R0), SYMBOLSTRING                   ; 1646
                                               0B  11 0002F           BRB     3$                                     ; 1641
                                52       09   A0  9A 00031  2$:      MOVZBL  9(R0), LENGTH                          ; 1650
                                52            0A  C0 00035           ADDL2   #10, LENGTH                            ;
                                63       09   A0  9E 00038           MOVAB   9(R0), SYMBOLSTRING                    ; 1652
                                08       02   A0  E8 0003C  3$:      BLBS    2(R0), 4$                              ; 1654
                         0000V  CF            00  FB 00040           CALLS   #0, PROSYMBOL                          ; 1656
                                07            50  E9 00045           BLBC    STATUS, 5$                             ;
                            FC  A3            52  C0 00048  4$:      ADDL2   LENGTH, GSDOFFSET                      ; 1658
                                50            01  D0 0004C           MOVL    #1, R0                                 ; 1659
                                                  04 0004F  5$:      RET                                            ; 1660

; Routine Size:  80 bytes,    Routine Base:  $CODE$ + 04A4
;  611           1661  1 %SBTTL  'entpnts';
;  612           1662  1
;  613           1663  1 ROUTINE entpnts =
;  614           1664  2 BEGIN
;  615           1665  2 !
;  616           1666  2 LOCAL
;  617           1667  2         length;
;  618           1668  2 BIND
;  619           1669  2         symbolrec = objvec [.gsdoffset] : BBLOCK;
;  620           1670  2
;  621           1671  2
;  622           1672  2 length = $BYTEOFFSET(epm$t_name) - $BYTEOFFSET(epm$t_start) +
;  623           1673  2                         .symbolrec [epm$b_namlng];
;  624           1674  2 symbolstring = symbolrec [epm$b_namlng];                              ! Point to the symbol
;  625           1675  2 perform (prosymbol ());
;  626           1676  2 gsdoffset = .gsdoffset + .length;                               ! Else update the offset for next
;  627           1677  2 RETURN true
;  628           1678  1 END;                                               ! Of entpnts



                                          000C 00000 ENTPNTS:.WORD    Save R2,R3                                          ; 1663
                       53     0000'    CF   9E 00002          MOVAB   GSDOFFSET, R3
              50     0C       A3             63   C1 00007          ADDL3   GSDOFFSET, OBJVEC, R0                               ; 1669
                       52        0B    A0   9A 0000C          MOVZBL  11(R0), LENGTH                                      ; 1672
                       52              0C   C0 00010          ADDL2   #12, LENGTH                                         ;
                  04   A3        0B    A0   9E 00013          MOVAB   11(R0), SYMBOLSTRING                                ; 1674
               0000V   CF              00   FB 00018          CALLS   #0, PROSYMBOL                                       ; 1675
                       06              50   E9 0001D          BLBC    STATUS, 1$                                          ;
                       63              52   C0 00020          ADDL2   LENGTH, GSDOFFSET                                   ; 1676
                       50              01   D0 00023          MOVL    #1, R0                                              ; 1677
                                            04 00026 1$:      RET                                                         ; 1678

; Routine Size:  39 bytes,    Routine Base:  $CODE$ + 04F4
;  630      1679 1 %SBTTL  'procedef';
;  631      1680 1
;  632      1681 1 ROUTINE procedef =
;  633      1682 2 BEGIN
;  634      1683 2 !
;  635      1684 2 !        A procedure definition is an extended entry point definition, carrying with
;  636      1685 2 !        it a description of the procedure's formal arguments. processing these consists
;  637      1686 2 !        in normal symbol definition processing followed by:-
;  638      1687 2 !                (1) Validation of the format of formal description (i.e. just check
;  639      1688 2 !                        that minimum number of arguments specified is less than
;  640      1689 2 !                        or equal to the maximum.
;  641      1690 2 !
;  642      1691 2 !
;  643      1692 2 LOCAL
;  644      1693 2     argcount;
;  645      1694 2
;  646      1695 2 IF .objvec [.gsdoffset] EQL obj$c_gsd_prow
;  647      1696 2 THEN
;  648      1697 3     perform (pro_epmw ())
;  649      1698 2 ELSE
;  650      1699 2     perform (entpnts ());
;  651      1700 2
;  652      1701 3 BEGIN
;  653      1702 3     BIND
;  654      1703 3         formals = objvec [.gsdoffset] : BBLOCK;
;  655      1704 3         gsdoffset = .gsdoffset + fml$c_size;                   ! Update record pointer
;  656      1705 3         IF (argcount = .formals [fml$b_maxargs]) NEQ 0         ! If args
;  657      1706 3         THEN INCRU i FROM 1 TO .argcount                       ! then skip them
;  658      1707 4         DO BEGIN
;  659      1708 4             BIND
;  660      1709 4                 argdesc = objvec [.gsdoffset] : BBLOCK;
;  661      1710 4
;  662      1711 4             gsdoffset = .gsdoffset + .argdesc [arg$b_bytecnt] + arg$c_size;
;  663      1712 3             END;
;  664      1713 3         RETURN true
;  665      1714 2       END;
;  666      1715 1 END;                                     ! Of procedef



                                        000C 00000 PROCEDEF:
                                                              .WORD   Save R2,R3                          ; 1681
                    53        0000'   CF  9E 00002            MOVAB   GSDOFFSET, R3                       ;
    50        0C    A3                63  C1 00007            ADDL3   GSDOFFSET, OBJVEC, R0               ; 1695
                    06                60  91 0000C            CMPB    (R0), #6                            ;
                    07                    12 0000F            BNEQ    1$                                  ;
          0000V     CF                00  FB 00011            CALLS   #0, PRO_EPMW                        ; 1697
                    04                    11 00016            BRB     2$                                  ;
              BD    AF                00  FB 00018 1$:        CALLS   #0, ENTPNTS                         ; 1699
                    2D                50  E9 0001C 2$:        BLBC    STATUS, 6$                          ;
    50        0C    A3                63  C1 0001F            ADDL3   GSDOFFSET, OBJVEC, R0               ; 1703
                    63                02  C0 00024            ADDL2   #2, GSDOFFSET                       ; 1704
                    52        01      A0  9A 00027            MOVZBL  1(R0), ARGCOUNT                     ; 1705
                    1C                    13 0002B            BEQL    5$                                  ;
                    51                01  D0 0002D            MOVL    #1, I                               ; 1706
```

```
                                            12  11 00030          BRB     4$                                  ;
                       50       0C    A3    63  C1 00032 3$:      ADDL3   GSDOFFSET, OBJVEC, R0              ; 1709
                                      50    01  A0  9A 00037      MOVZBL  1(R0), R0                          ; 1711
                                      50        63  C0 0003B      ADDL2   GSDOFFSET, R0                      ;
                                      63    02  A0  9E 0003E      MOVAB   2(R0), GSDOFFSET                   ;
                                                51  D6 00042      INCL    I                                  ; 1706
                                      52        51  D1 00044 4$:  CMPL    I, ARGCOUNT                        ;
                                                E9  1B 00047      BLEQU   3$                                 ;
                                      50        01  D0 00049 5$:  MOVL    #1, R0                             ; 1713
                                                    04 0004C 6$:  RET                                        ; 1715

; Routine Size:  77 bytes,    Routine Base:  $CODE$ + 051B
;   668          1716  1 %SBTTL  'pro_epmw';
;   669          1717  1
;   670          1718  1 ROUTINE pro_epmw =
;   671          1719  2 BEGIN
;   672          1720  2 !
;   673          1721  2 !        Process entry points with word psect
;   674          1722  2 !
;   675          1723  2 LOCAL
;   676          1724  2          length;
;   677          1725  2 BIND
;   678          1726  2          symbolrec = objvec [.gsdoffset] : BBLOCK;
;   679          1727  2
;   680          1728  2
;   681          1729  2 length = $BYTEOFFSET(epmw$t_name) - $BYTEOFFSET(epmw$t_start) +
;   682          1730  2                     .symbolrec [epmw$b_namlng];
;   683          1731  2 symbolstring = symbolrec [epmw$b_namlng];                           ! Point to the symbol
;   684          1732  2 perform (prosymbol ());
;   685          1733  2 gsdoffset = .gsdoffset + .length;                           ! Else update the offset for next
;   686          1734  2 RETURN true
;   687          1735  1 END;                                                 ! Of pro_epmw



                                                     000C 00000 PRO_EPMW:
                                                                          .WORD    Save R2,R3                                    ; 1718
                               53      0000'  CF   9E 00002          MOVAB    GSDOFFSET, R3
                 50      0C    A3         63   C1 00007          ADDL3    GSDOFFSET, OBJVEC, R0                         ; 1726
                               52        0C   A0   9A 0000C          MOVZBL   12(R0), LENGTH                                ; 1729
                               52             0D   C0 00010          ADDL2    #13, LENGTH                                   ;
                         04    A3        0C   A0   9E 00013          MOVAB    12(R0), SYMBOLSTRING                          ; 1731
                       0000V   CF             00   FB 00018          CALLS    #0, PROSYMBOL                                 ; 1732
                               06             50   E9 0001D          BLBC     STATUS, 1$                                    ;
                               63             52   C0 00020          ADDL2    LENGTH, GSDOFFSET                             ; 1733
                               50             01   D0 00023          MOVL     #1, R0                                        ; 1734
                                                   04 00026 1$:      RET                                                    ; 1735

; Routine Size:  39 bytes,    Routine Base:  $CODE$ + 0568


;   688          1736  1
;  690          1737  1 %SBTTL  'pro_idc';
;  691          1738  1
;  692          1739  1 ROUTINE pro_idc =
;  693          1740  2 BEGIN
;  694          1741  2 !
;  695          1742  2 !       Process random entity check
;  696          1743  2 !       by skipping it.
;  697          1744  2 !
;  698          1745  2 LOCAL
;  699          1746  2     identstring : REF VECTOR [,BYTE],    ! pointer to ident string
;  700          1747  2     objectname : REF VECTOR [,BYTE],     ! pointer to object name string
;  701          1748  2     length;
;  702          1749  2 BIND
;  703          1750  2         idc_rec = objvec [.gsdoffset] : BBLOCK;
;  704          1751  2
;  705          1752  2 identstring = idc_rec [idc$b_namlng] + 1 + .idc_rec [idc$b_namlng];
;  706          1753  2 objectname = identstring [1] + .identstring [0];
;  707          1754  2 length =  objectname [1] + .objectname [0] - idc_rec;
;  708          1755  2 gsdoffset = .gsdoffset + .length;
;  709          1756  2 RETURN true
;  710          1757  1 END;                                              ! Of pro_idc



                                               0004 00000 PRO_IDC:.WORD   Save R2                                       ; 1739
                 52     0000'  CF      0000'  CF  C1 00002         ADDL3   GSDOFFSET, OBJVEC, R2                        ; 1750
                               50         03  A2  9A 0000A         MOVZBL  3(R2), R0                                    ; 1752
                               50      04 A042    9E 0000E         MOVAB   4(R0)[R2], IDENTSTRING                       ;
                               51             60  9A 00013         MOVZBL  (IDENTSTRING), R1                            ; 1753
                               50      01 A140    9E 00016         MOVAB   1(R1)[IDENTSTRING], OBJECTNAME               ;
                               51             60  9A 0001B         MOVZBL  (OBJECTNAME), R1                             ; 1754
                               50             51  C0 0001E         ADDL2   R1, OBJECTNAME                               ;
                               50             52  C2 00021         SUBL2   R2, R0                                       ;
                                              50  D6 00024         INCL    LENGTH                                       ;
                        0000'  CF             50  C0 00026         ADDL2   LENGTH, GSDOFFSET                            ; 1755
                               50             01  D0 0002B         MOVL    #1, R0                                       ; 1756
                                                  04 0002E         RET                                                  ; 1757

; Routine Size:  47 bytes,    Routine Base:  $CODE$ + 058F


;  711          1758  1
;  713          1759  1 %SBTTL  'pro_env';
;  714          1760  1
;  715          1761  1 ROUTINE pro_env =
;  716          1762  2 BEGIN
;  717          1763  2 !
;  718          1764  2 !       Process environment definition
;  719          1765  2 !       by skipping it.
;  720          1766  2 !
;  721          1767  2 LOCAL
;  722          1768  2         length;
;  723          1769  2 BIND
;  724          1770  2         env_rec = objvec [.gsdoffset] : BBLOCK;
;  725          1771  2
;  726          1772  2
;  727          1773  2 length =  env_rec [env$t_name] - objvec [.gsdoffset] +
;  728          1774  2                         .env_rec [env$b_namlng];
;  729          1775  2 gsdoffset = .gsdoffset + .length;
;  730          1776  2 RETURN true
;  731          1777  1 END;                                            ! Of pro_env



                                            0004 00000 PRO_ENV:.WORD    Save R2                                              ; 1761
                     50     0000'  CF     0000'  CF  C1 00002         ADDL3   GSDOFFSET, OBJVEC, R0                          ; 1770
                     51            50            50  C3 0000A         SUBL3   R0, R0, R1                                     ; 1773
                                   52        05  A0  9A 0000E         MOVZBL  5(R0), R2                                      ; 1774
                                   51            52  C0 00012         ADDL2   R2, R1                                         ;
                                   50        06  A1  9E 00015         MOVAB   6(R1), LENGTH                                  ; 1773
                            0000'  CF            50  C0 00019         ADDL2   LENGTH, GSDOFFSET                              ; 1775
                                   50            01  D0 0001E         MOVL    #1, R0                                         ; 1776
                                                     04 00021         RET                                                    ; 1777

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 05BE


;  732          1778  1
;  734          1779  1 %SBTTL  'pro_lsy';
;  735          1780  1
;  736          1781  1 ROUTINE pro_lsy =
;  737          1782  2 BEGIN
;  738          1783  2 !
;  739          1784  2 !       Process local symbol definition/reference
;  740          1785  2 !       by skipping it.
;  741          1786  2 !
;  742          1787  2 LOCAL
;  743          1788  2         length;
;  744          1789  2 BIND
;  745          1790  2         lsy_rec = objvec [.gsdoffset] : BBLOCK;
;  746          1791  2
;  747          1792  2 IF NOT .lsy_rec [lsy$v_def]
;  748          1793  2 THEN
;  749          1794  2     length = $BYTEOFFSET(lsrf$t_name) - $BYTEOFFSET(lsrf$t_start) +
;  750          1795  2                         .lsy_rec [lsrf$b_namlng]
;  751          1796  2 ELSE
;  752          1797  2     length = $BYTEOFFSET(lsdf$t_name) - $BYTEOFFSET(lsdf$t_start) +
;  753          1798  2                         .lsy_rec [lsdf$b_namlng];
;  754          1799  2 gsdoffset = .gsdoffset + .length;
;  755          1800  2 RETURN true
;  756          1801  1 END;                                                     ! Of pro_lsy



                                          0000 00000 PRO_LSY:.WORD    Save nothing                                 ; 1781
            50     0000'  CF     0000'  CF  C1 00002          ADDL3    GSDOFFSET, OBJVEC, R0                        ; 1790
            09        02  A0           01  E0 0000A          BBS      #1, 2(R0), 1$                                ; 1792
                          50       06  A0  9A 0000F          MOVZBL   6(R0), LENGTH                                ; 1794
                          50           07  C0 00013          ADDL2    #7, LENGTH                                   ;
                                       07  11 00016          BRB      2$                                           ;
                          50       0C  A0  9A 00018 1$:      MOVZBL   12(R0), LENGTH                               ; 1797
                          50           0D  C0 0001C          ADDL2    #13, LENGTH                                  ;
                   0000'  CF           50  C0 0001F 2$:      ADDL2    LENGTH, GSDOFFSET                            ; 1799
                          50           01  D0 00024          MOVL     #1, R0                                       ; 1800
                                           04 00027          RET                                                   ; 1801

; Routine Size:  40 bytes,    Routine Base:  $CODE$ + 05E0


;  757          1802  1
;  759          1803  1 %SBTTL  'pro_lepm';
;  760          1804  1
;  761          1805  1 ROUTINE pro_lepm =
;  762          1806  2 BEGIN
;  763          1807  2 !
;  764          1808  2 !       Process local symbol entry point definition
;  765          1809  2 !       by skipping it.
;  766          1810  2 !
;  767          1811  2 LOCAL
;  768          1812  2         length;
;  769          1813  2 BIND
;  770          1814  2         lepm_rec = objvec [.gsdoffset] : BBLOCK;
;  771          1815  2
;  772          1816  2
;  773          1817  2 length = $BYTEOFFSET(lepm$t_name) - $BYTEOFFSET(lepm$t_start) +
;  774          1818  2                         .lepm_rec [lepm$b_namlng];
;  775          1819  2 gsdoffset = .gsdoffset + .length;                              ! Else update the offset for next
;  776          1820  2 RETURN true
;  777          1821  1 END;                                              ! Of pro_lepm




                                              0000 00000 PRO_LEPM:
                                                                .WORD   Save nothing                                    ; 1805
                  50      0000'  CF     0000'  CF  C1 00002    ADDL3   GSDOFFSET, OBJVEC, R0                          ; 1814
                                 50        0E  A0  9A 0000A    MOVZBL  14(R0), LENGTH                                 ; 1817
                                 50            0F  C0 0000E    ADDL2   #15, LENGTH                                    ;
                          0000'  CF            50  C0 00011    ADDL2   LENGTH, GSDOFFSET                              ; 1819
                                 50            01  D0 00016    MOVL    #1, R0                                         ; 1820
                                                   04 00019    RET                                                    ; 1821

; Routine Size:  26 bytes,    Routine Base:  $CODE$ + 0608


;  778          1822  1
;  780          1823  1 %SBTTL  'pro_lpro';
;  781          1824  1
;  782          1825  1 ROUTINE pro_lpro =
;  783          1826  2 BEGIN
;  784          1827  2 !
;  785          1828  2 !       Process local symbol procedure definition
;  786          1829  2 !       by skipping it.
;  787          1830  2 !
;  788          1831  2 LOCAL
;  789          1832  2         length;
;  790          1833  2 BIND
;  791          1834  2         lpro_rec = objvec [.gsdoffset] : BBLOCK;
;  792          1835  2
;  793          1836  2
;  794          1837  2 length = $BYTEOFFSET(lpro$t_name) - $BYTEOFFSET(lpro$t_start) +
;  795          1838  2                       .lpro_rec [lpro$b_namlng];
;  796          1839  2 gsdoffset = .gsdoffset + .length;                           ! Else update the offset for next
;  797          1840  2 RETURN true
;  798          1841  1 END;                                              ! Of pro_lpro



                                              0000 00000 PRO_LPRO:
                                                              .WORD   Save nothing                                   ; 1825
                  50     0000'  CF      0000'  CF  C1 00002   ADDL3   GSDOFFSET, OBJVEC, R0                          ; 1834
                                50        0E   A0  9A 0000A   MOVZBL  14(R0), LENGTH                                 ; 1837
                                50             0F  C0 0000E   ADDL2   #15, LENGTH                                    ;
                         0000'  CF             50  C0 00011   ADDL2   LENGTH, GSDOFFSET                              ; 1839
                                50             01  D0 00016   MOVL    #1, R0                                         ; 1840
                                                   04 00019   RET                                                    ; 1841

; Routine Size:  26 bytes,    Routine Base:  $CODE$ + 0622


;  799          1842  1
;   801         1843    1 %SBTTL  'pro_spsc';
;   802         1844    1
;   803         1845    1 ROUTINE pro_spsc =
;   804         1846    2 BEGIN
;   805         1847    2 !
;   806         1848    2 !       Process shareable image psect definition
;   807         1849    2 !       by ignoring it.
;   808         1850    2 !
;   809         1851    2 LOCAL
;   810         1852    2         length;
;   811         1853    2 BIND
;   812         1854    2         spsct_def = objvec [.gsdoffset] : BBLOCK;
;   813         1855    2
;   814         1856    2 !
;   815         1857    2 !       First check for legal P-section name and alignment
;   816         1858    2 !
;   817         1859    2 IF .spsct_def [sgps$b_namlng] GTRU sym$c_maxlng              ! Check name within the legal
;   818         1860    2 OR .spsct_def [sgps$b_namlng] EQL 0                          ! Range for symbol and P-section
;   819         1861    3 THEN BEGIN
;   820         1862    3     SIGNAL (lib$_spnamlng, 3, modnamlng, lib$gl_inpfdb [fdb$l_namdesc],
;   821         1863    3                 .spsct_def [sgps$b_namlng]);
;   822         1864    3     RETURN lib$_spnamlng;
;   823         1865    2     END;
;   824         1866    2
;   825         1867    2 length = $BYTEOFFSET(sgps$t_name) - $BYTEOFFSET(sgps$t_start) +
;   826         1868    2                         .spsct_def [sgps$b_namlng];
;   827         1869    2 gsdoffset = .gsdoffset + .length;
;   828         1870    2 RETURN true
;   829         1871    1 END;                                          ! Of pro_spsc



                                         001C 00000 PRO_SPSC:
                                                              .WORD   Save R2,R3,R4                          ; 1845
                      54      0000'   CF  9E 00002            MOVAB   GSDOFFSET, R4                          ;
                      53 00000000G    8F  D0 00007            MOVL    #LIB$_SPNAMLNG, R3                     ;
       52       0C    A4              64  C1 0000E            ADDL3   GSDOFFSET, OBJVEC, R2                  ; 1854
                      1F        0C    A2  91 00013            CMPB    12(R2), #31                            ; 1859
                                      05  1A 00017            BGTRU   1$                                     ;
                                0C    A2  95 00019            TSTB    12(R2)                                 ; 1860
                                      1C  12 0001C            BNEQ    2$                                     ;
                      7E        0C    A2  9A 0001E 1$:        MOVZBL  12(R2), -(SP)                          ; 1863
       7E    0000G    CF              10  C1 00022            ADDL3   #16, LIB$GL_INPFDB, -(SP)              ; 1862
                                1C    A4  9F 00028            PUSHAB  MODNAMLNG                              ;
                                      03  DD 0002B            PUSHL   #3                                     ;
                                      53  DD 0002D            PUSHL   R3                                     ;
          00000000G   00              05  FB 0002F            CALLS   #5, LIB$SIGNAL                         ;
                      50              53  D0 00036            MOVL    R3, R0                                 ; 1864
                                          04 00039            RET                                            ;
                      50        0C    A2  9A 0003A 2$:        MOVZBL  12(R2), LENGTH                         ; 1867
                      50              0D  C0 0003E            ADDL2   #13, LENGTH                            ;
                      64              50  C0 00041            ADDL2   LENGTH, GSDOFFSET                      ; 1869
                      50              01  D0 00044            MOVL    #1, R0                                 ; 1870
                                          04 00047            RET                                            ; 1871
```

```
; Routine Size:  72 bytes,    Routine Base:  $CODE$ + 063C

;  830          1872  1
```

```
  1 %SBTTL  'prosymbol';
  1
  1 ROUTINE prosymbol =
  2 BEGIN
  2 !++
  2 !
  2 !
  2 !--
  2 IF .symbolstring [0] GTRU .lib$gl_keysize                          ! If the symbol length is outside
  2 OR .symbolstring [0] EQL 0                                          ! Legal range
  3 THEN BEGIN
  3     SIGNAL (lib$_symnamlng, 4, symbolstring [0], modnamlng,
  3             lib$gl_inpfdb [fdb$l_namdesc], .symbolstring [0]);
  3     RETURN lib$_symnamlng;
  2     END;
  2 IF NOT .lib$gl_ctlmsk [lib$v_globals]
  2 THEN RETURN true
  3 ELSE BEGIN
  3
  3     LOCAL
  3         status,
  3         replacekey,
  3         keynb : REF BBLOCK,
  3         txtrfa : BBLOCK [rfa$c_length],
  3         keydesc : BBLOCK [dsc$c_s_bln];
  3
  3     keydesc [dsc$w_length] = .symbolstring [0];
  3     keydesc [dsc$a_pointer] = symbolstring [1];
P 3     perform (lbr$set_index (lib$gl_libctl, lib$gl_objgsdix),
  3                     lib$_indexerr, 1, lib$gl_libfdb [fdb$l_namdesc]);
  3 !
  3 !     If the symbol is already in the index and we are not replacing, then that is
  3 !     an error.  If we are replacing, it must be from the same module, otherwise
  3 !     that is an error.
  3 !
  4     IF (replacekey = lbr$lookup_key (lib$gl_libctl, keydesc, txtrfa))   !If key already in index
  4         AND (IF .lib$gl_ctlmsk [lib$v_replace]
  4                     THEN NOT CH$EQL (rfa$c_length, txtrfa, rfa$c_length, oldmodrfa)
  4                     ELSE true)
  4     THEN BEGIN
  4         SIGNAL (lib$_dupglobal, 3, keydesc, lib$gl_inpfdb [fdb$l_namdesc], !Tell user of error
  4                             lib$gl_libfdb [fdb$l_namdesc]);
  4         RETURN lib$_dupglobal;
  3         END;
  3 !
  3 ! If replacing the key, look and see if its on the deleted key list.  If it is, remove it
  3 ! from that list, and put on the global list.  If not replacing, just put on the global
  3 ! list.
  3 !
  3     status = false;
  4     IF NOT (
  4             IF .replacekey
  5             THEN BEGIN
  5                 keynb = delist [0];                                 !Initialize to search queue
  5                 WHILE (keynb = .keynb [lnb$l_flink]) NEQ delist [0]
  5                     DO IF CH$EQL (.keydesc [dsc$w_length], .keydesc [dsc$a_pointer],
  5                                 .keynb [lnb$b_namlng], keynb [lnb$t_name])
;  889        1930  6                          THEN BEGIN
;  890        1931  6                               REMQUE (.keynb, keynb);           !Remove from the deleted symbol queue
;  891        1932  6                               status = true;
;  892        1933  6                               EXITLOOP;
;  893        1934  5                               END;
;  894        1935  4                     END;
;  895        1936  4                     .status                                     !Result of search
;  896        1937  4               )
;  897        1938  3       THEN
;  898        1939  4           BEGIN
;  899        1940  4           LOCAL
;  900        1941  4               key_nb : REF BBLOCK;                                ! search globlist to be sure symbol not already on l
;  901        1942  4
;  902        1943  4           key_nb = globlist [0];
;  903        1944  4           WHILE (key_nb = .key_nb [lnb$l_flink]) NEQ globlist [0] DO
;  904        1945  4               IF CH$EQL (.keydesc [dsc$w_length], .keydesc [dsc$a_pointer],
;  905        1946  4                           .key_nb [lnb$b_namlng], key_nb [lnb$t_name])
;  906        1947  4               THEN RETURN true;                                   ! Key already in list, so exist
;  907        1948  4           perform (lib_get_mem (lnb$c_fixedsize + .keydesc [dsc$w_length], keynb));
;  908        1949  4           keynb [lnb$b_namlng] = .keydesc [dsc$w_length];
;  909        1950  4           CH$MOVE (.keydesc [dsc$w_length], .keydesc [dsc$a_pointer], keynb [lnb$t_name]);
;  910        1951  3           END;
;  911        1952  3       keynb [lnb$v_replace] = .replacekey;
;  912        1953  3       INSQUE (.keynb, .globlist [1]);
;  913        1954  2       END;
;  914        1955  2 RETURN true
;  915        1956  1 END;                                                              ! Of symbol



                                                   07FC 00000 PROSYMBOL:
                                                                      .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10              ; 1875
                        5A 00000000G  8F  D0 00002                    MOVL    #LIB$_DUPGLOBAL, R10                          ;
                        59 00000000G  8F  D0 00009                    MOVL    #LIB$_SYMNAMLNG, R9                           ;
                        58 00000000G  00  9E 00010                    MOVAB   LIB$SIGNAL, R8                                ;
                        57     0000'  CF  9E 00017                    MOVAB   SYMBOLSTRING, R7                              ;
                        5E            14  C2 0001C                    SUBL2   #20, SP                                       ;
                        50       00   B7  9A 0001F                    MOVZBL  @SYMBOLSTRING, R0                             ; 1881
               0000G    CF            50  D1 00023                    CMPL    R0, LIB$GL_KEYSIZE                            ;
                                      04  1A 00028                    BGTRU   1$                                            ;
                                      50  D5 0002A                    TSTL    R0                                            ; 1882
                                      18  12 0002C                    BNEQ    2$                                            ;
                                      50  DD 0002E  1$:               PUSHL   R0                                            ; 1885
         7E    0000G    CF            10  C1 00030                    ADDL3   #16, LIB$GL_INPFDB, -(SP)                     ;
                                 18   A7  9F 00036                    PUSHAB  MODNAMLNG                                     ; 1884
                                      67  DD 00039                    PUSHL   SYMBOLSTRING                                  ; 1885
                                      04  DD 0003B                    PUSHL   #4                                            ;
                                      59  DD 0003D                    PUSHL   R9                                            ;
                        68            06  FB 0003F                    CALLS   #6, LIB$SIGNAL                                ;
                        50            59  D0 00042                    MOVL    R9, R0                                        ; 1886
                                          04 00045                    RET                                                   ;
         03    0000G    CF            01  E0 00046  2$:               BBS     #1, LIB$GL_CTLMSK+2, 3$                       ; 1888
                                    00F2  31 0004C                    BRW     12$                                           ;
                  04    AE       00   B7  9B 0004F  3$:               MOVZBW  @SYMBOLSTRING, KEYDESC                        ; 1899
      08 AE             67            01  C1 00054                    ADDL3   #1, SYMBOLSTRING, KEYDESC+4                   ; 1900
```

```
                                                       E 14
LIB_INPUTOBJ                                           16-Sep-1984 01:57:57    VAX-11 Bliss-32 V4.0-742          Page  39
V04-000          prosymbol                             14-Sep-1984 12:38:04    [LIBRAR.SRC]INPUTOBJ.B32;1             (20)

                                          0000G  CF  9F 00059         PUSHAB  LIB$GL_OBJGSDIX                           ; 1902
                                          0000G  CF  9F 0005D         PUSHAB  LIB$GL_LIBCTL                             ;
                       00000000G  00             02  FB 00061         CALLS   #2, LBR$SET_INDEX                         ;
                                  13             50  E8 00068         BLBS    STATUS, 4$                                ;
                                                 50  DD 0006B         PUSHL   STATUS                                    ;
                   7E      0000G  CF             10  C1 0006D         ADDL3   #16, LIB$GL_LIBFDB, -(SP)                 ;
                                                 01  DD 00073         PUSHL   #1                                        ;
                                      00000000G  8F  DD 00075         PUSHL   #LIB$_INDEXERR                            ;
                                  68             04  FB 0007B         CALLS   #4, LIB$SIGNAL                            ;
                                             0C  AE  9F 0007E 4$:     PUSHAB  TXTRFA                                    ; 1908
                                             08  AE  9F 00081         PUSHAB  KEYDESC                                   ;
                                          0000G  CF  9F 00084         PUSHAB  LIB$GL_LIBCTL                             ;
                       00000000G  00             03  FB 00088         CALLS   #3, LBR$LOOKUP_KEY                        ;
                                  56             50  D0 0008F         MOVL    R0, REPLACEKEY                            ;
                                  28             56  E9 00092         BLBC    REPLACEKEY, 6$                            ;
                   08      0000G  CF             05  E1 00095         BBC     #5, LIB$GL_CTLMSK+1, 5$                   ; 1909
             40    A7         0C  AE             06  29 0009B         CMPC3   #6, TXTRFA, OLDMODRFA                     ; 1910
                                                 1A  13 000A1         BEQL    6$                                        ;
                   7E      0000G  CF             10  C1 000A3 5$:     ADDL3   #16, LIB$GL_LIBFDB, -(SP)                 ; 1914
                   7E      0000G  CF             10  C1 000A9         ADDL3   #16, LIB$GL_INPFDB, -(SP)                 ; 1913
                                             0C  AE  9F 000AF         PUSHAB  KEYDESC                                   ;
                                                 03  DD 000B2         PUSHL   #3                                        ; 1914
                                                 5A  DD 000B4         PUSHL   R10                                       ;
                                  68             05  FB 000B6         CALLS   #5, LIB$SIGNAL                            ;
                                  50             5A  D0 000B9         MOVL    R10, R0                                   ; 1915
                                                     04 000BC         RET                                               ;
                                                 55  D4 000BD 6$:     CLRL    STATUS                                    ; 1922
                                  2E             56  E9 000BF         BLBC    REPLACEKEY, 8$                            ; 1924
                                  6E       0080  C7  9E 000C2         MOVAB   DELIST, KEYNB                             ; 1926
                                  50         00  BE  D0 000C7 7$:     MOVL    @KEYNB, R0                                ; 1927
                                  6E             50  D0 000CB         MOVL    R0, KEYNB                                 ;
                                  51       0080  C7  9E 000CE         MOVAB   DELIST, R1                                ;
                                  51             50  D1 000D3         CMPL    R0, R1                                    ;
                                                 18  13 000D6         BEQL    8$                                        ;
                                  54             6E  D0 000D8         MOVL    KEYNB, R4                                 ; 1929
                                  50         09  A4  9A 000DB         MOVZBL  9(R4), R0                                 ;
      50           00         08  BE         04  AE  2D 000DF         CMPC5   KEYDESC, @KEYDESC+4, #0, R0, 10(R4)       ;
                                             0A  A4     000E6                                                           ;
                                                 DD  12 000E8         BNEQ    7$                                        ;
                                  6E             64  0F 000EA         REMQUE  (R4), KEYNB                               ; 1931
                                  55             01  D0 000ED         MOVL    #1, STATUS                                ; 1932
                                  41             55  E8 000F0 8$:     BLBS    STATUS, 11$                               ; 1936
                                  54         78  A7  9E 000F3         MOVAB   GLOBLIST, KEY_NB                          ; 1943
                                  54             64  D0 000F7 9$:     MOVL    (KEY_NB), KEY_NB                          ; 1944
                                  50         78  A7  9E 000FA         MOVAB   GLOBLIST, R0                              ;
                                  50             54  D1 000FE         CMPL    KEY_NB, R0                                ;
                                                 11  13 00101         BEQL    10$                                       ;
                                  50         09  A4  9A 00103         MOVZBL  9(KEY_NB), R0                             ; 1946
      50           00         08  BE         04  AE  2D 00107         CMPC5   KEYDESC, @KEYDESC+4, #0, R0, 10(KEY_NB)   ;
                                             0A  A4     0010E                                                           ;
                                                 E5  12 00110         BNEQ    9$                                        ;
                                                 2D  11 00112         BRB     12$                                       ; 1947
                                                 5E  DD 00114 10$:    PUSHL   SP                                        ; 1948
                                  7E         08  AE  3C 00116         MOVZWL  KEYDESC, -(SP)                            ;
                                  6E             0A  C0 0011A         ADDL2   #10, (SP)                                 ;
                           0000G  CF             02  FB 0011D         CALLS   #2, LIB_GET_MEM                           ;
                                  1F             50  E9 00122         BLBC    STATUS, 13$                               ;
```

```
                                                                          F 14
LIB_INPUTOBJ                                                        16-Sep-1984 01:57:57    VAX-11 Bliss-32 V4.0-742              Page  40
V04-000              prosymbol                                      14-Sep-1984 12:38:04    [LIBRAR.SRC]INPUTOBJ.B32;1                 (20)

                                     50            6E  D0 00125          MOVL    KEYNB, R0                                       ; 1949
                               09    A0      04    AE  90 00128          MOVB    KEYDESC, 9(R0)                                  ;
                   0A    A0    08    BE      04    AE  28 0012D          MOVC3   KEYDESC, @KEYDESC+4, 10(R0)                     ; 1950
                                     50            6E  D0 00134 11$:     MOVL    KEYNB, R0                                       ; 1952
     08    A0            01          00            56  F0 00137          INSV    REPLACEKEY, #0, #1, 8(R0)                       ;
                               7C    B7            60  0E 0013D          INSQUE  (R0), @GLOBLIST+4                               ; 1953
                                     50            01  D0 00141 12$:     MOVL    #1, R0                                          ; 1955
                                                       04 00144 13$:     RET                                                     ; 1956

; Routine Size:  325 bytes,    Routine Base:  $CODE$ + 0684
```

```
  1 %SBTTL  'proeom';
  1
  1 ROUTINE proeom =
  2 BEGIN
  2 !
  2 !       Process end of module records:
  2 !               (1) Validate sequence
  2 !               (2) Interpret compiler completion code,
  2 !                   issuing appropriate error or warning message
  2 !
  2
  2 LOCAL
  2         datadesc : BBLOCK [dsc$c_s_bln],
  2         modnamdesc : BBLOCK [dsc$c_s_bln],
  2         comcode;
  2 !
  2 maxreclng = obj$c_maxrecsiz;                                      !Reset max record length
  2 perform (seqchk ());
  2 IF (comcode = .objrec [eom$b_comcod]) NEQ 0                       ! If non zero compilation cplete code
  3 THEN BEGIN                                                        ! CHECK
  3     IF .comcode GTRU 3 THEN comcode = 4;                          !Make illegal index legal
  3     IF .comcode NEQ 0
  3     THEN SIGNAL (lib$_comcod, 3, compilecods [.comcode * dsc$c_s_bln,0,0,0], !Signal the error (warning)
  3                 modnamlng, lib$gl_inpfdb [fdb$l_namdesc]);
  3     END;
  2 perform (copyrec ());
P 2 rms_perform (lbr$put_end (lib$gl_libctl),
  2         lib$_writeerr, .lbr$gl_rmsstv, 1, lib$gl_libfdb [fdb$l_namdesc]);
  2 !
  2 ! Update the module header
  2 !
  2 IF .lib$gl_ctlmsk [lib$v_selective]
  2 THEN moduleflags = .moduleflags OR mhd$m_selsrc;
  2 datadesc [dsc$w_length] = .idlng + 2;                     !include flag and id length bytes
  2 datadesc [dsc$a_pointer] = moduleflags;
  2 modnamdesc [dsc$w_length] = .modnamlng;
  2 modnamdesc [dsc$a_pointer] = modulename;
P 2 rms_perform (lbr$set_module (lib$gl_libctl, modulerfa ,0,0, datadesc),
  2         lib$_mhderr, .lbr$gl_rmsstv, 2, modnamdesc, lib$gl_libfdb [fdb$l_namdesc]);
  2 !
  2 ! Insert all the keys now
  2 !
  2 perform (finish_object (true));
  2 !
  2 ! Log operation if logging on console
  2 !
  2 lib_log_upd (
  2     (IF .operation EQL lib$_replaced THEN lhe$c_replaced ELSE lhe$c_inserted),
  2     modnamdesc ); ! log module name for LUH record
  2 lib_log_op (.operation, modnamdesc, .lib$gl_libfdb); !Log insert if /LOG
  2 modulerfa [rfa$l_vbn] = 0;                                !Reset module VBN address
  2 globlist [0] = globlist [0];
  2 globlist [1] = globlist [0];
  2 moduleflags = 0;
  2 modnamlng = 0;
  2 RETURN true
  1 END;                            ! END OF EOM PROCESSING
```

```
                                    003C 00000 PROEOM: .WORD    Save R2,R3,R4,R5                             ; 1959
              55      0000G   CF   9E 00002            MOVAB    LIB$GL_LIBFDB, R5                            ;
              54  00000000G   00   9E 00007            MOVAB    LBR$GL_RMSSTV, R4                            ;
              53  00000000G   00   9E 0000E            MOVAB    LIB$SIGNAL, R3                               ;
              52      0000'   CF   9E 00015            MOVAB    MODNAMLNG, R2                                ;
              5E              10   C2 0001A            SUBL2    #16, SP                                      ;
       FC     A2      0800    8F   3C 0001D            MOVZWL   #2048, MAXRECLNG                             ; 1973
    0000V     CF              00   FB 00023            CALLS    #0, SEQCHK                                   ; 1974
              32              50   E9 00028            BLBC     STATUS, 3$                                   ;
              50        F0    A2   D0 0002B            MOVL     OBJREC, R0                                   ; 1975
              50        01    A0   9A 0002F            MOVZBL   1(R0), COMCODE                               ;
                              23   13 00033            BEQL     2$                                           ;
              03              50   D1 00035            CMPL     COMCODE, #3                                  ; 1977
                              03   1B 00038            BLEQU    1$                                           ;
              50              04   D0 0003A            MOVL     #4, COMCODE                                  ;
                              50   D5 0003D 1$:        TSTL     COMCODE                                      ; 1978
                              17   13 0003F            BEQL     2$                                           ;
 7E       0000G   CF              10   C1 00041            ADDL3    #16, LIB$GL_INPFDB, -(SP)                    ; 1980
                              52   DD 00047            PUSHL    R2                                           ; 1979
                        70 A240   7F 00049            PUSHAQ   COMPILECODS[COMCODE]                         ;
                              03   DD 0004D            PUSHL    #3                                           ; 1980
                  00000000G   8F   DD 0004F            PUSHL    #LIB$_COMCOD                                 ;
              63              05   FB 00055            CALLS    #5, LIB$SIGNAL                               ;
    0000V     CF              00   FB 00058 2$:        CALLS    #0, COPYREC                                  ; 1982
              74              50   E9 0005D 3$:        BLBC     STATUS, 7$                                   ;
                      0000G   CF   9F 00060            PUSHAB   LIB$GL_LIBCTL                                ; 1984
00000000G     00              01   FB 00064            CALLS    #1, LBR$PUT_END                              ;
              13              50   E8 0006B            BLBS     STATUS, 4$                                   ;
                              64   DD 0006E            PUSHL    LBR$GL_RMSSTV                                ;
                              50   DD 00070            PUSHL    STATUS                                       ;
 7E                   65      10   C1 00072            ADDL3    #16, LIB$GL_LIBFDB, -(SP)                    ;
                              01   DD 00076            PUSHL    #1                                           ;
                  008610D2    8F   DD 00078            PUSHL    #8786130                                     ;
              63              05   FB 0007E            CALLS    #5, LIB$SIGNAL                               ;
 04       0000G   CF              02   E1 00081 4$:        BBC      #2, LIB$GL_CTLMSK+2, 5$                      ; 1988
              3C     A2              01   88 00087            BISB2    #1, MODULEFLAGS                              ; 1989
              08     AE        3D    A2   9B 0008B 5$:        MOVZBW   IDLNG, DATADESC                              ; 1990
              08     AE              02   A0 00090            ADDW2    #2, DATADESC                                 ;
              0C     AE        3C    A2   9E 00094            MOVAB    MODULEFLAGS, DATADESC+4                      ; 1991
              6E                     62   9B 00099            MOVZBW   MODNAMLNG, MODNAMDESC                        ; 1992
              04     AE        01    A2   9E 0009C            MOVAB    MODULENAME, MODNAMDESC+4                     ; 1993
                               08    AE   9F 000A1            PUSHAB   DATADESC                                     ; 1995
                                     7E   7C 000A4            CLRQ     -(SP)                                        ;
                               20    A2   9F 000A6            PUSHAB   MODULERFA                                    ;
                           0000G     CF   9F 000A9            PUSHAB   LIB$GL_LIBCTL                                ;
00000000G     00                     05   FB 000AD            CALLS    #5, LBR$SET_MODULE                           ;
              16                     50   E8 000B4            BLBS     STATUS, 6$                                   ;
                                     64   DD 000B7            PUSHL    LBR$GL_RMSSTV                                ;
                                     50   DD 000B9            PUSHL    STATUS                                       ;
 7E                   65             10   C1 000BB            ADDL3    #16, LIB$GL_LIBFDB, -(SP)                    ;
                               0C    AE   9F 000BF            PUSHAB   MODNAMDESC                                   ;
                                     02   DD 000C2            PUSHL    #2                                           ;
```

```
LIB_INPUTOBJ                                                    I 14
V04-000         proeom                                 16-Sep-1984 01:57:57    VAX-11 Bliss-32 V4.0-742                Page  43
                                                       14-Sep-1984 12:38:04    [LIBRAR.SRC]INPUTOBJ.B32;1                   (21)

                                000000000G  8F    DD 000C4          PUSHL   #LIB$_MHDERR                              ;
                                63          06    FB 000CA          CALLS   #6, LIB$SIGNAL                            ;
                                            01    DD 000CD 6$:      PUSHL   #1                                        ; 1999
                      0000V  CF             01    FB 000CF          CALLS   #1, FINISH_OBJECT                         ;
                             39             50    E9 000D4 7$:      BLBC    STATUS, 10$                               ;
                                            5E    DD 000D7          PUSHL   SP                                        ; 2003
                 00000000G   8F       D4    A2    D1 000D9          CMPL    OPERATION, #LIB$_REPLACED                 ; 2004
                                            04    12 000E1          BNEQ    8$                                        ;
                                            03    DD 000E3          PUSHL   #3                                        ;
                                            02    11 000E5          BRB     9$                                        ;
                                            02    DD 000E7 8$:      PUSHL   #2                                        ;
                      0000G  CF             02    FB 000E9 9$:      CALLS   #2, LIB_LOG_UPD                           ;
                                            65    DD 000EE          PUSHL   LIB$GL_LIBFDB                             ; 2006
                                      04    AE    9F 000F0          PUSHAB  MODNAMDESC                                ;
                                      D4    A2    DD 000F3          PUSHL   OPERATION                                 ;
                      0000G  CF             03    FB 000F6          CALLS   #3, LIB_LOG_OP                            ;
                                      20    A2    D4 000FB          CLRL    MODULERFA                                 ; 2007
                        60   A2       60    A2    9E 000FE          MOVAB   GLOBLIST, GLOBLIST                        ; 2008
                        64   A2       60    A2    9E 00103          MOVAB   GLOBLIST, GLOBLIST+4                      ; 2009
                                      3C    A2    94 00108          CLRB    MODULEFLAGS                               ; 2010
                                            62    94 0010B          CLRB    MODNAMLNG                                 ; 2011
                             50             01    D0 0010D          MOVL    #1, R0                                    ; 2012
                                                  04 00110 10$:     RET                                               ; 2013

; Routine Size:  273 bytes,    Routine Base:  $CODE$ + 07C9
```

```
  1 %SBTTL  'finish_object';
  1
  1 ROUTINE finish_object (allswell) =
  2 BEGIN
  2 !
  2 !       This routine is called when the processing for a module is complete.
  2 !       if allswell is true, the symbols in the queue and the module name
  2 !       are entered in the index, and the old data and any symbols not replaced
  2 !       (if replacing) are deleted from the index.  If allswell is false,
  2 !       the list is merely deallocated.
  2 !
  2 LOCAL
  2     keydesc : BBLOCK [dsc$c_s_bln],
  2     keynb : REF BBLOCK;
  2
  2
  2 !
  2 ! Write the end of the data if there was an error and then delete it
  2 !
  2 IF .modulerfa [rfa$l_vbn] NEQ 0                         !If data was written
  2     AND NOT .allswell                                   ! and there was an error
  3 THEN BEGIN
  3     lbr$put_end (lib$gl_libctl);
  3     lbr$delete_data (lib$gl_libctl, modulerfa);         !Delete the new data
  3     modulerfa [rfa$l_vbn] = 0;
  2     END;
  2 !
  2 ! Set index to the global symbol index
  2 !
P 2 perform (lbr$set_index (lib$gl_libctl, lib$gl_objgsdix),
  2                         lib$_indexerr, 1, lib$gl_libfdb [fdb$l_namdesc]);
  2 !
  2 ! Enter the new symbols
  2 !
  2 WHILE NOT REMQUE (.globlist, keynb)                                 !Insert/replace symbols for module
  3 DO BEGIN
  3     IF .allswell
  4     THEN BEGIN
  4         keydesc [dsc$w_length] = .keynb [lnb$b_namlng];
  4         keydesc [dsc$a_pointer] = keynb [lnb$t_name];
P 4         rms_perform (lbr$replace_key (lib$gl_libctl, keydesc,
P 4                                 oldmodrfa, modulerfa),
P 4                         lib$_inserterr, .lbr$gl_rmsstv,
  4                         2, keydesc, lib$gl_libfdb [fdb$l_namdesc]);
  3         END;
  3     lib_free_mem (lnb$c_fixedsize + .keynb [lnb$b_namlng], .keynb);
  2     END;
  2 !
  2 ! Delete any symbols not replaced
  2 !
  2 WHILE NOT REMQUE (.delist, keynb)
  3 DO BEGIN
  3     IF .allswell
  4     THEN BEGIN
  4         keydesc [dsc$w_length] = .keynb [lnb$b_namlng];
  4         keydesc [dsc$a_pointer] = keynb [lnb$t_name];
P 4         perform (lbr$delete_key (lib$gl_libctl, keydesc),
; 1032          2071 4                      lib$_delkeyerr, 2, keydesc, lib$gl_libfdb [fdb$l_namdesc]);
; 1033          2072 3              END;
; 1034          2073 3          lib_free_mem (lnb$c_fixedsize + .keynb [lnb$b_naming], .keynb);
; 1035          2074 2          END;
; 1036          2075 2  IF .allswell
; 1037          2076 3  THEN BEGIN
; 1038        P 2077 3      perform (lbr$set_index (lib$gl_libctl, lib$gl_objmodix),
; 1039          2078 3                          lib$_indexerr, 1, lib$gl_libfdb [fdb$l_namdesc]);
; 1040        P 2079 3      rms_perform (lbr$replace_key (lib$gl_libctl, moduledesc,
; 1041        P 2080 3                                  oldmodrfa, modulerfa),
; 1042        P 2081 3                          lib$_inserterr, .lbr$gl_rmsstv,
; 1043          2082 3                          2, moduledesc, lib$gl_libfdb [fdb$l_namdesc]);
; 1044          2083 3  !
; 1045          2084 3  ! If replacing, delete the old data
; 1046          2085 3  !
; 1047          2086 3      IF .replacing
; 1048        P 2087 3      THEN rms_perform (lbr$delete_data (lib$gl_libctl, oldmodrfa),
; 1049          2088 3                  lib$_deldaterr, .lbr$gl_rmsstv, 1, lib$gl_libfdb [fdb$l_namdesc]);
; 1050          2089 2      END;
; 1051          2090 2
; 1052          2091 2  RETURN true
; 1053          2092 1  END;                                                          !Of deallocate_list



                                          0FFC 00000 FINISH_OBJECT:
                                                             .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 2016
                       5B 00000000G   00   9E 00002          MOVAB   LBR$REPLACE_KEY, R11                          ;
                       5A 00000000G   8F   D0 00009          MOVL    #LIB$_INDEXERR, R10                           ;
                       59 00000000G   00   9E 00010          MOVAB   LBR$SET_INDEX, R9                             ;
                       58 00000000G   00   9E 00017          MOVAB   LBR$DELETE_DATA, R8                           ;
                       57 00000000G   00   9E 0001E          MOVAB   LBR$GL_RMSSTV, R7                             ;
                       56     0000G   CF   9E 00025          MOVAB   LIB$GL_LIBFDB, R6                             ;
                       55     0000G   CF   9E 0002A          MOVAB   LIB$GL_LIBCTL, R5                             ;
                       54 00000000G   00   9E 0002F          MOVAB   LIB$SIGNAL, R4                                ;
                       53     0000'   CF   9E 00036          MOVAB   MODULERFA, R3                                 ;
                       5E             08   C2 0003B          SUBL2   #8, SP                                        ;
                                      63   D5 0003E          TSTL    MODULERFA                                     ; 2033
                                      16   13 00040          BEQL    1$                                            ;
                       12        04   AC   E8 00042          BLBS    ALLSWELL, 1$                                  ; 2034
                                      55   DD 00046          PUSHL   R5                                            ; 2036
            00000000G  00             01   FB 00048          CALLS   #1, LBR$PUT_END                               ;
                                      53   DD 0004F          PUSHL   R3                                            ; 2037
                                      55   DD 00051          PUSHL   R5                                            ;
                       68             02   FB 00053          CALLS   #2, LBR$DELETE_DATA                           ;
                                      63   D4 00056          CLRL    MODULERFA                                     ; 2038
                              0000G   CF   9F 00058 1$:      PUSHAB  LIB$GL_OBJGSDIX                               ; 2044
                                      55   DD 0005C          PUSHL   R5                                            ;
                       69             02   FB 0005E          CALLS   #2, LBR$SET_INDEX                             ;
                       0D             50   E8 00061          BLBS    STATUS, 2$                                    ;
                                      50   DD 00064          PUSHL   STATUS                                        ;
         7E            66             10   C1 00066          ADDL3   #16, LIB$GL_LIBFDB, -(SP)                     ;
                                      01   DD 0006A          PUSHL   #1                                            ;
                                      5A   DD 0006C          PUSHL   R10                                           ;
                       64             04   FB 0006E          CALLS   #4, LIB$SIGNAL                                ;
```

```
                    52          40    B3  0F 00071 2$:    REMQUE  @GLOBLIST, KEYNB                          ; 2048
                                      43  1D 00075        BVS     4$                                        ;
                    2F          04    AC  E9 00077        BLBC    ALLSWELL, 3$                              ; 2050
                    6E          09    A2  9B 0007B        MOVZBW  9(KEYNB), KEYDESC                         ; 2052
            04      AE          0A    A2  9E 0007F        MOVAB   10(R2), KEYDESC+4                         ; 2053
                                      53  DD 00084        PUSHL   R3                                        ; 2057
                                08    A3  9F 00086        PUSHAB  OLDMODRFA                                 ;
                                08    AE  9F 00089        PUSHAB  KEYDESC                                   ;
                                      55  DD 0008C        PUSHL   R5                                        ;
                    6B                04  FB 0008E        CALLS   #4, LBR$REPLACE_KEY                       ;
                    16                50  E8 00091        BLBS    STATUS, 3$                                ;
                                      67  DD 00094        PUSHL   LBR$GL_RMSSTV                             ;
                                      50  DD 00096        PUSHL   STATUS                                    ;
        7E          66                10  C1 00098        ADDL3   #16, LIB$GL_LIBFDB, -(SP)                 ;
                                0C    AE  9F 0009C        PUSHAB  KEYDESC                                   ;
                                      02  DD 0009F        PUSHL   #2                                        ;
                         00000000G    8F  DD 000A1        PUSHL   #LIB$_INSERTERR                           ;
                    64                06  FB 000A7        CALLS   #6, LIB$SIGNAL                            ;
                                      52  DD 000AA 3$:    PUSHL   KEYNB                                     ; 2059
                    7E          09    A2  9A 000AC        MOVZBL  9(KEYNB), -(SP)                           ;
                    6E                0A  C0 000B0        ADDL2   #10, (SP)                                 ;
          0000G     CF                02  FB 000B3        CALLS   #2, LIB_FREE_MEM                          ;
                                      B7  11 000B8        BRB     2$                                        ; 2048
                    52          48    B3  0F 000BA 4$:    REMQUE  @DELIST, KEYNB                            ; 2064
                                      3F  1D 000BE        BVS     6$                                        ;
                    2B          04    AC  E9 000C0        BLBC    ALLSWELL, 5$                              ; 2066
                    6E          09    A2  9B 000C4        MOVZBW  9(KEYNB), KEYDESC                         ; 2068
            04      AE          0A    A2  9E 000C8        MOVAB   10(R2), KEYDESC+4                         ; 2069
                              4020    8F  BB 000CD        PUSHR   #^M<R5,SP>                                ; 2071
      00000000G     00                02  FB 000D1        CALLS   #2, LBR$DELETE_KEY                        ;
                    14                50  E8 000D8        BLBS    STATUS, 5$                                ;
                                      50  DD 000DB        PUSHL   STATUS                                    ;
        7E          66                10  C1 000DD        ADDL3   #16, LIB$GL_LIBFDB, -(SP)                 ;
                                08    AE  9F 000E1        PUSHAB  KEYDESC                                   ;
                                      02  DD 000E4        PUSHL   #2                                        ;
                         00000000G    8F  DD 000E6        PUSHL   #LIB$_DELKEYERR                           ;
                    64                05  FB 000EC        CALLS   #5, LIB$SIGNAL                            ;
                                      52  DD 000EF 5$:    PUSHL   KEYNB                                     ; 2073
                    7E          09    A2  9A 000F1        MOVZBL  9(KEYNB), -(SP)                           ;
                    6E                0A  C0 000F5        ADDL2   #10, (SP)                                 ;
          0000G     CF                02  FB 000F8        CALLS   #2, LIB_FREE_MEM                          ;
                                      BB  11 000FD        BRB     4$                                        ; 2064
                    61          04    AC  E9 000FF 6$:    BLBC    ALLSWELL, 9$                              ; 2075
                             0000G    CF  9F 00103        PUSHAB  LIB$GL_OBJMODIX                           ; 2078
                                      55  DD 00107        PUSHL   R5                                        ;
                    69                02  FB 00109        CALLS   #2, LBR$SET_INDEX                         ;
                    0D                50  E8 0010C        BLBS    STATUS, 7$                                ;
                                      50  DD 0010F        PUSHL   STATUS                                    ;
        7E          66                10  C1 00111        ADDL3   #16, LIB$GL_LIBFDB, -(SP)                 ;
                                      01  DD 00115        PUSHL   #1                                        ;
                                      5A  DD 00117        PUSHL   R10                                       ;
                    64                04  FB 00119        CALLS   #4, LIB$SIGNAL                            ;
                                      53  DD 0011C 7$:    PUSHL   R3                                        ; 2082
                                08    A3  9F 0011E        PUSHAB  OLDMODRFA                                 ;
                                14    A3  9F 00121        PUSHAB  MODULEDESC                                ;
                                      55  DD 00124        PUSHL   R5                                        ;
                    6B                04  FB 00126        CALLS   #4, LBR$REPLACE_KEY                       ;
```

```
                              16               50  E8 00129          BLBS    STATUS, 8$                                    ;
                                               67  DD 0012C          PUSHL   LBR$GL_RMSSTV                                 ;
                                               50  DD 0012E          PUSHL   STATUS                                        ;
                7E            66               10  C1 00130          ADDL3   #16, LIB$GL_LIBFDB, -(SP)                     ;
                                        14     A3  9F 00134          PUSHAB  MODULEDESC                                    ;
                                               02  DD 00137          PUSHL   #2                                            ;
                                 00000000G     8F  DD 00139          PUSHL   #LIB$_INSERTERR                               ;
                              64               06  FB 0013F          CALLS   #6, LIB$SIGNAL                                ;
                              1E        10     A3  E9 00142 8$:      BLBC    REPLACING, 9$                                 ; 2086
                                        08     A3  9F 00146          PUSHAB  OLDMODRFA                                     ; 2088
                                               55  DD 00149          PUSHL   R5                                            ;
                              68               02  FB 0014B          CALLS   #2, LBR$DELETE_DATA                           ;
                              13               50  E8 0014E          BLBS    STATUS, 9$                                    ;
                                               67  DD 00151          PUSHL   LBR$GL_RMSSTV                                 ;
                                               50  DD 00153          PUSHL   STATUS                                        ;
                7E            66               10  C1 00155          ADDL3   #16, LIB$GL_LIBFDB, -(SP)                     ;
                                               01  DD 00159          PUSHL   #1                                            ;
                                 00000000G     8F  DD 0015B          PUSHL   #LIB$_DELDATERR                               ;
                              64               05  FB 00161          CALLS   #5, LIB$SIGNAL                                ;
                              50               01  D0 00164 9$:      MOVL    #1, R0                                        ; 2091
                                                   04 00167          RET                                                   ; 2092

; Routine Size:  360 bytes,    Routine Base:  $CODE$ + 08DA
; 1055          2093  1 %SBTTL  'seqchk';
; 1056          2094  1
; 1057          2095  1 ROUTINE seqchk =
; 1058          2096  1 !
; 1059          2097  1 !       Routine which validates that records are in correct sequence.
; 1060          2098  1 !       Returns value false if not, true otherwise.
; 1061          2099  1 !
; 1062          2100  2 BEGIN
; 1063          2101  2 BIND
; 1064          2102  2         hdrsubtyp = objrec [obj$b_subtyp] : BYTE;
; 1065          2103  2 !
; 1066          2104  2 IF .currectyp EQL obj$c_hdr
; 1067          2105  2 THEN                                                   !If this record is a header
; 1068          2106  2         IF .hdrsubtyp EQL obj$c_hdr_mhd                !and it is the main module header
; 1069          2107  2         THEN                                           !then we have valid sequence
; 1070          2108  2                 IF (.lastrectyp EQL obj$c_eom) OR      !if and only if the previous
; 1071          2109  3                    (.lastrectyp EQL obj$c_eomw)
; 1072          2110  3                 THEN (mhdseen = true;                  !Record was end of module.  if that
; 1073          2111  3                       lnmseen = false;                 !
; 1074          2112  3                    RETURN true)                        !is the case set mhd record
; 1075          2113  3                 ELSE BEGIN
; 1076          2114  3                     SIGNAL (lib$_seqnce, 2, modnamlng,
; 1077          2115  3                             [ib$gl_inpfdb [fdb$l_namdesc]);
; 1078          2116  3                     RETURN lib$_seqnce;
; 1079          2117  3                     END
; 1080          2118  2         ELSE
; 1081          2119  2                 IF .mhdseen                            !If some other kind of header
; 1082          2120  3                 THEN (IF .hdrsubtyp EQL obj$c_hdr_lnm  !we must have seen a main header
; 1083          2121  3                       THEN lnmseen = true;
; 1084          2122  3                       RETURN true)
; 1085          2123  3                 ELSE BEGIN
; 1086          2124  3                     SIGNAL (lib$_seqnce, 2, modnamlng,
; 1087          2125  3                             [ib$gl_inpfdb [fdb$l_namdesc]);
; 1088          2126  3                     RETURN lib$_seqnce;
; 1089          2127  3                     END
; 1090          2128  2 ELSE
; 1091          2129  2         IF .mhdseen
; 1092          2130  2         AND .lnmseen
; 1093          2131  2         THEN                                           !If we have seen a main header
; 1094          2132  3                 BEGIN
; 1095          2133  3                 IF (.currectyp EQL obj$c_eom) OR       !then turn off flag on end of module.
; 1096          2134  4                    (.currectyp EQL obj$c_eomw)
; 1097          2135  3                 THEN mhdseen = false;                  !sequence error if have not seen
; 1098          2136  3                 RETURN true;                           !main header and this is not one.
; 1099          2137  3                 END
; 1100          2138  3         ELSE BEGIN
; 1101          2139  3             SIGNAL (lib$_seqnce, 2, modnamlng,
; 1102          2140  3                         lib$gl_inpfdb [fdb$l_namdesc]);
; 1103          2141  3             RETURN lib$_seqnce;
; 1104          2142  2             END;
; 1105          2143  1 END;



                                  000C 00000 SEQCHK: .WORD   Save R2,R3                                          ; 2095
```

```
                                                         B 15
LIB_INPUTOBJ                                     16-Sep-1984 01:57:57    VAX-11 Bliss-32 V4.0-742          Page  49
V04-000          seqchk                          14-Sep-1984 12:38:04    [LIBRAR.SRC]INPUTOBJ.B32;1            (23)

                          53 00000000G   8F  D0 00002          MOVL    #LIB$_SEQNCE, R3                     ;
                          52     0000'   CF  9E 00009          MOVAB   MHDSEEN, R2                          ;
          51        18    A2             01  C1 0000E          ADDL3   #1, OBJREC, R1                       ; 2102
                          50        20   A2  D0 00013          MOVL    CURRECTYP, R0                        ; 2104
                                         23  12 00017          BNEQ    3$                                   ;
                                         61  95 00019          TSTB    (R1)                                 ; 2106
                                         11  12 0001B          BNEQ    2$                                   ;
                          03        1C   A2  D1 0001D          CMPL    LASTRECTYP, #3                       ; 2108
                                         06  13 00021          BEQL    1$                                   ;
                          07        1C   A2  D1 00023          CMPL    LASTRECTYP, #7                       ; 2109
                                         2A  12 00027          BNEQ    6$                                   ;
                          62             01  7D 00029 1$:      MOVQ    #1, MHDSEEN                          ; 2110
                                         21  11 0002C          BRB     5$                                   ; 2112
                          22             62  E9 0002E 2$:      BLBC    MHDSEEN, 6$                          ; 2119
                          01             61  91 00031          CMPB    (R1), #1                             ; 2120
                                         19  12 00034          BNEQ    5$                                   ;
                    04    A2             01  D0 00036          MOVL    #1, LNMSEEN                          ; 2121
                                         13  11 0003A          BRB     5$                                   ; 2122
                          14             62  E9 0003C 3$:      BLBC    MHDSEEN, 6$                          ; 2129
                          10        04   A2  E9 0003F          BLBC    LNMSEEN, 6$                          ; 2130
                          03             50  D1 00043          CMPL    R0, #3                               ; 2133
                                         05  13 00046          BEQL    4$                                   ;
                          07             50  D1 00048          CMPL    R0, #7                               ; 2134
                                         02  12 0004B          BNEQ    5$                                   ;
                                         62  D4 0004D 4$:      CLRL    MHDSEEN                              ; 2135
                          50             01  D0 0004F 5$:      MOVL    #1, R0                               ; 2136
                                             04 00052          RET                                          ;
          7E     0000G    CF             10  C1 00053 6$:      ADDL3   #16, LIB$GL_INPFDB, -(SP)            ; 2140
                                    28   A2  9F 00059          PUSHAB  MODNAMLNG                            ; 2139
                                         02  DD 0005C          PUSHL   #2                                   ; 2140
                                         53  DD 0005E          PUSHL   R3                                   ;
             00000000G    00             04  FB 00060          CALLS   #4, LIB$SIGNAL                       ;
                          50             53  D0 00067          MOVL    R3, R0                               ; 2141
                                             04 0006A          RET                                          ; 2143

; Routine Size:  107 bytes,    Routine Base:  $CODE$ + 0A42
; 1107          2144  1 %SBTTL  'prorec';
; 1108          2145  1
; 1109          2146  1 ROUTINE prorec =
; 1110          2147  2 BEGIN
; 1111          2148  2 !
; 1112          2149  2 ! This routine checks for proper record sequence and then
; 1113          2150  2 ! copies the record to the object library.
; 1114          2151  2 !
; 1115          2152  2 perform (seqchk ());                               !Check sequence
; 1116          2153  2 IF NOT .lib$gl_ctlmsk [lib$v_shrstb]
; 1117          2154  2     THEN RETURN copyrec ()                         !Copy to library
; 1118          2155  2     ELSE RETURN true
; 1119          2156  1 END;                              !Of prorec



                                        0000 00000 PROREC: .WORD    Save nothing                                    ; 2146
                 8F    AF                 00  FB 00002          CALLS    #0, SEQCHK                                      ; 2152
                 0F                       50  E9 00006          BLBC     STATUS, 2$                                      ;
           06        0000G  CF            05  E0 00009          BBS      #5, LIB$GL_CTLMSK, 1$                           ; 2153
                     0000V  CF            00  FB 0000F          CALLS    #0, COPYREC                                     ; 2154
                                              04 00014          RET                                                      ; 2155
                 50                       01  D0 00015 1$:      MOVL     #1, R0                                          ;
                                              04 00018 2$:      RET                                                      ; 2156

; Routine Size:  25 bytes,    Routine Base:  $CODE$ + 0AAD


; 1120          2157  1 ROUTINE copyrec =
; 1121          2158  2 BEGIN
; 1122          2159  2 !
; 1123          2160  2 ! This routine copies the record to the object library
; 1124          2161  2 !
; 1125          2162  2 LOCAL
; 1126          2163  2         txtrfa : BBLOCK [rfa$c_length],
; 1127          2164  2         bufdesc : BBLOCK [dsc$c_s_bln];
; 1128          2165  2
; 1129          2166  2 bufdesc [dsc$w_length] = .reclng;
; 1130          2167  2 bufdesc [dsc$a_pointer] = .objrec;
; 1131        P 2168  2 rms_perform (lbr$put_record  (lib$gl_libctl, bufdesc, txtrfa),
; 1132          2169  2         lib$_writeerr, .lbr$gl_rmsstv, 1, lib$gl_libfdb [fdb$l_namdesc]);
; 1133          2170  2 IF .modulerfa [rfa$l_vbn] EQL 0
; 1134          2171  3 THEN BEGIN
; 1135          2172  3     modulerfa [rfa$l_vbn] = .txtrfa [rfa$l_vbn];
; 1136          2173  3     modulerfa [rfa$w_offset] = .txtrfa [rfa$w_offset];
; 1137          2174  2     END;
; 1138          2175  2 RETURN true
; 1139          2176  1 END;                              !Of copyrec



                                        0004 00000 COPYREC:.WORD    Save R2                                         ; 2157
                 52     0000'  CF   9E 00002          MOVAB    MODULERFA, R2                                   ;
```

```
                          5E                10  C2 00007          SUBL2   #16, SP                              ;
                          6E          CC    A2  B0 0000A          MOVW    RECLNG, BUFDESC                      ; 2166
                   04     AE          D0    A2  D0 0000E          MOVL    OBJREC, BUFDESC+4                    ; 2167
                                      08    AE  9F 00013          PUSHAB  TXTRFA                               ; 2169
                                      04    AE  9F 00016          PUSHAB  BUFDESC                              ;
                                   0000G    CF  9F 00019          PUSHAB  LIB$GL_LIBCTL                        ;
               00000000G  00                03  FB 0001D          CALLS   #3, LBR$PUT_RECORD                   ;
                          1D                50  E8 00024          BLBS    STATUS, 1$                           ;
                           00000000G        00  DD 00027          PUSHL   LBR$GL_RMSSTV                        ;
                                            50  DD 0002D          PUSHL   STATUS                               ;
          7E      0000G   CF                10  C1 0002F          ADDL3   #16, LIB$GL_LIBFDB, -(SP)            ;
                                            01  DD 00035          PUSHL   #1                                   ;
                           008610D2         8F  DD 00037          PUSHL   #8786130                             ;
               00000000G  00                05  FB 0003D          CALLS   #5, LIB$SIGNAL                       ;
                                            62  D5 00044 1$:      TSTL    MODULERFA                            ; 2170
                                            09  12 00046          BNEQ    2$                                   ;
                          62          08    AE  D0 00048          MOVL    TXTRFA, MODULERFA                    ; 2172
                   04     A2          0C    AE  B0 0004C          MOVW    TXTRFA+4, MODULERFA+4                ; 2173
                          50                01  D0 00051 2$:      MOVL    #1, R0                               ; 2175
                                                04 00054          RET                                          ; 2176

; Routine Size:  85 bytes,    Routine Base:  $CODE$ + 0AC6


; 1140          2177  1
; 1141          2178  1 END
; 1142          2179  0 ELUDOM



                                                                  .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY
;
;       Name                     Bytes                         Attributes
;
;   $OWN$                          200  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                         152  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                        2843  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                             Library Statistics
;
;                                       -------- Symbols --------       Pages       Processing
;       File                            Total     Loaded   Percent      Mapped      Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1       18619       120         0       1000          00:01.9
;                                      COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:INPUTOBJ/OBJ=OBJ$:INPUTOBJ MSRC$:INPUTOBJ/UPDATE=(ENH$:INPUTOBJ)

; Size:          2843 code + 352 data bytes
; Run Time:         00:56.3
; Elapsed Time:     02:02.7
; Lines/CPU Min:     2321
; Lexemes/CPU-Min: 28165
; Memory Used:  275 pages
; Compilation Complete
```

0201 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY